Université Nice Sophia Antipolis

Habilitation Thesis
Habilitation à Diriger des Recherches (HDR)

Major: Computer Science

Catherine Faron Zucker

June 12th, 2017

Knowledge Modelling and Processing for the Social Semantic Web

Jury:

Nathalie Aussenac Gilles,	Research Director,	CNRS, France,	Reviewer
Fabien Gandon,	Research Director,	INRIA, France	
Jean-Marc Labat,	Professor,	UPMC, France	
Marie-Christine Rousset,	Professor,	UGA, France,	Reviewer
Harald Sack,	Professor,	KIT, Germany,	Reviewer

to Rose,
to Alice,
to Léo,

Acknowledgments

Jean-Gabriel Ganascia who introduced me to AI when I was a master student and who supervised my PhD thesis.
Rose Dieng who offered me a post-doctoral position in the ACACIA team at Inria Sophia Antipolis and who made me decide to have an academic career. I admired her and I loved her.
My long-standing colleagues from the ACACIA days: Alain Giboin, the senior of the team; Olivier Corby, my favorite co-author according to dblp; and Fabien Gandon. Fabien is not only one of my oldest and favorite colleagues; he became my closest one when I became vice-head of Wimmics. I esteem his scientific vision, I enjoy our daily interactions, and I am very grateful to him for his contagious energy and for his strong support until my habilitation defense.
Peter Sander, who recruited me in the MAINLINE team at the I3S laboratory of the University Nice Sophia Antipolis, as a post-doctoral researcher and then as an assistant professor.
Nhan Le Thanh and Michel Riveill who first encouraged me several years ago to defend my habilitation thesis.
Andrea Tettamanzi and Johan Montagnat with whom I started more recently significant and enjoyable collaborations.
Olivier Corby, Oscar Rodriguez and Andrea Tettamanzi who very kindly took on my teaching load and pedagogical responsibilities this year to give me the time necessary to write my habilitation thesis.
Igor Litovsky for his cynical humour and some discussions that helped me step back while writing my habilitation thesis.
Hélène Collavizza for sharing her yogic science and strongly encouraging me and supporting me over these last years to defend my habilitation thesis.
My colleagues from the board of the French society for Artificial Intelligence and the French community on Knowledge Engineering who sent me many friendly "pep talks" about writing my habilitation thesis.
Last but not least: all my present and past master students, PhD students, and young researchers that I supervised.
All in all, the research period reported in this habilitation thesis and its writing itself were a tremendous human adventure!

Contents

List of Figures

List of Tables

Listings

Introduction

This document relates and synthesizes my research and research management experience since I joined the ACACIA team led by Rose Dieng Kuntz in 2000 for a postdoctoral position at Inria Sophia Antipolis Méditerranée (Inria SAM). Then I joined in 2001 the MAINLINE team led by Peter Sander for a postdoctoral position at the I3S laboratory of the University Nice Sophia Antipolis where I got an Assistant Professor position (*Maître de Conférences*) in 2002. In 2007, the MAINLINE team was restructured into the KEWI (Knowledge Engineering and Web Intelligence) team, led by Nhan Le Than and I became its vice-head. In the Meantime, the ACACIA team had become the EDELWEISS team led by Rose Dieng Kuntz and then by Olivier Corby. Then EDELWEISS and KEWI merged to become WIMMICS (Web Instrumented Man-Machine Interactions, Communities and Semantics) in 2012, a joint team between Inria, University Nice Sophia Antipolis and CNRS. It is led by Fabien Gandon and I am its vice-head. WIMMICS is a sub-group of the SPARKS team (Scalable and Pervasive softwARe and Knowledge Systems) in I3S which has been structured into three themes in 2015 and I am coordinating one of them with Alain Giboin: FORUM (FORmalising and Reasoning with Users and Models).

Throughout this 16-year period, my research activities have taken place in the domain of the semantic Web. My background was a PhD in Artificial Intelligence, and more specifically in Knowledge Representation and Reasoning (KRR). During my PhD, directed by Jean-Gabriel Ganascia at the University of Paris 6, I worked on the representation of taxonomic knowledge in the Conceptual Graph formalism and on the construction of hypermedias based on such formalization. The principles underlying this work were in direct line with those of the emerging semantic Web — annotation of resources in a graph-based model and reasoning on these formal representations—, with the difference that the Hypercard hypermedia system considered at that time was limited to a single machine while the Web spans a world-wide network. My research topics gradually evolved from representing and reasoning on digital resources to representing and reasoning on social ecosystems where not only digital resources are modelled but also users, users' knowledge and users' activities, thus bridging the gap between formal semantics and social semantics. This is what is reported in this document.

Application Domains and Research Projects

The main application domains of my research work are e-Education and Organisational and Collective Memories.

E-Education

Regarding my activities in the e-Education domain, I was involved in the European project TRIAL SOLUTION in 2001 and in the French project WebLearn supported by CNRS and Inria in 2003-2004; in the continuation of these projects, I participated to the supervision of both the PhD of Sylvain Dehors directed by Rose Dieng Kuntz on the exploitation of semantic Web and Knowledge Management Technologies for E-learning [42] and the PhD of Amel Yessad directed by M.T. Laskri on the adaptation of navigation paths in an e-learning environment to learner profiles [95]. I was the French leader of a scientific cooperation with Algeria on collaborative ITS, supported by CNRS, from 2009 to 2012; in this framework I co-supervised, with my colleague Hassina Seridi from the University of Annaba, the PhD thesis of Samia Beldjoudi on the recommandation of pedagogical ressources adapted to user profiles [3]. In 2014 I initiated a collaboration with the French company GAYATECH with the EDUCLOUD project, supported by the ANR (Carnot project); I supervised the post-doctoral work of Oscar Rodriguez on the ontology-based modelling of a serious game with the aim of recommending adapted pedagogical resources. In the continuation of this collaboration, we are now working on the automatic generation of quizzes based on domain ontologies and Linked Data. Also in 2014, I initiated a collaboration with the French company Educlever and we obtained in 2016 the funding of the EduMICS joint laboratory with Inria on the semantic annotation of the pedagogical resources produced by the company, with the aim of integrating heterogeneous knowledge sources and semantically handle them. Since 2017, I supervise the post-doctoral work of Géraud Fokou on this project. At the beginning of 2017, the French project SIDES 3.0 funded by the ANR was launched that aims at the creation of an intelligent digital training environment carried by all faculties of medicine in France and I am the scientific leader of it for Inria SAM. I will supervise a post-doctoral researcher on the ontology-based modelling of the training environment, the ontology-based integration and standardization of heterogeneous data and the construction of semantic-intensive learning services. Finally, since 2014 I participate to the French network ORPHEE supported by the ANR agency and gathering French research actors in e-Education, and since 2016 I am the scientific referent of the Inria Learning Lab gathering the Inria researchers involved in the e-Education domain.

Organisational and collective memories

Regarding my activities in the domain of Organisational and Collective Memories, I was involved from 2006 to 20013 in a long-term collaboration with CSTB, the French scientific and technical institute for the Building industry, supported by CSTB, on assisting the exploitation of technical and regulatory documents; in this context, I co-supervised with my colleague Nhan Le Thanh the Phd thesis of Anastasiya Yurchyshyna on supporting conformity checking in the building industry [98] and the PhD thesis of Khalil Bouzidi on supporting the redaction and control of technical and regulatory documents [5]. Related to these works but in another domain, I was involved in 2009 in the French project DESIR supported by Inria on the capitalization of know-hows of agronomists and geneticists at INRA, the French research institute on agronomy. From 2009 to 2012, I participated to the French project ISICIL supported by the ANR agency

on the study and experiment with the usage of new tools relying on Web 2.0 advanced interfaces and semantic Web technologies to assist tasks of corporate intelligence and technical watch. From 2013 to 2016 I participated to the French project Ocktopus supported by the ANR agency on social network analysis for finding key agents and relevant answers to questions in question-answer sites or forum; this was the subject of the PhD thesis of Zide Meng that I co-supevised with my colleague Fabien Gandon [70]. Also in 2013, I initiated a collaboration with the French company SynchroNext, on the development of an ontology-based intelligent chatbot for commercial site support; this is the subject of the PhD thesis of Raphaël Gazzotti that I am co-supervising with Fabien Gandon. Finally, in 2015 I initiated a collaboration with the French company SILEX on modelling a social network of service providers and contractors and supporting interactions for the recommendation of providers. I just started co-supervising the PhD thesis of Molka Dhouib on this subject with Andrea Tettamanzi.

Cultural and scientific heritage

More recently, I started applying my work in the domain of Digital Humanities and more specifically in Cultural and Scientific Heritage. I participated to the creation of the international network Zoomathia on the study of the formation and transmission of ancient zoological knowledge in Antiquity and Middle Age. It is supported by CNRS since 2014 and I am the scientific leader for I3S. Also from 2014 to 2015, I was involved in the French-Italian project LIENS on the creation and publication of a semantic repository representing a collection of artworks owned by two Italian museums. Finally, from 2015 to 2016 I participated to the French AZKAR project supported by BPI and I was involved in the creation and publication of a semantic repository describing museum archives and historical scenes of the World War One museum in Meaux.

My scientific results in all the above-cited running or completed projects directly feed Chapter 1 and Chapter 2. Chapter 3 and Chapter 4 synthesizes the results of more theoretical works which have also been indirectly supported by these research projects.

Domain-independant projects

I am involved in a long-term collaborative work on graph-based knowledge representation with my colleagues Olivier Corby and Fabien Gandon, transversal to, and indirectly funded by the above cited research projects. On this specific topic, we were together involved in the French project GRIWES funded by Inria in 2008. More recently, we also co-supervised the PhD thesis of Oumy Seye funded by Inria on representing and reasoning on and with rules on the semantic Web [83]. From 2012 to 2015, I was involved in the French CrEDIBLE project supported by CNRS (MASTODONS programme) on federating distributed data sources; in this context I participated to the supervision of the PhD thesis of Alban Gaignard with my colleague Johan Montagnat on the semantic distribution of data sources [58], and I co-supervised with him the PhD thesis of Franck Michel on the federation of heterogenous data sources [75]. More recently I

started collaborating with colleague Andrea Tettamanzi on ontology learning from RDF data, in the continuation of my past work with Alexandre Delteil in the framework of his PhD thesis [47]. We are in the process of answering call for projects to support this research activity.

The rest of this document is organized as follows: Chapter 1 presents my works on modelling and reasoning on formal representations of digital resources and Chapter 2 presents my works on modelling users, community members and communities. Chapter 3 presents my works on graph-based KRR on the semantic Web, focusing on the formal semantics of the Web. Chapter 4 presents my works on developing foundational solutions for Linked Data management.

Chapter 1

Modelling and Managing Digital Resources in Epistemic Communities

Introduction

This chapter synthesizes my scientific contributions related to the management of digital resources of epistemic communities, addressing the general research question of *How should we model the social semantics of the digital resources shared within a community in order to efficiently manage them?* Going back to the origins of the semantic Web in the late 90s, the challenge was to enable a *semantic* and therefore *intelligent* retrieval of digital information resources or an intelligent navigation among them, over the Web and Intrawebs. This was in line with the current trends in the Knowledge Engineering research area, aiming at building knowledge-intensive systems. It supposed to somehow relate digital information resources gathered in a dataset, in a system, and the formal representation of the knowledge relative to the *domain* they belong to. This can be viewed as bridging the gap between the formal semantics of digital information resources gathered for some purpose(s) by the members of an online community and their formal semantics captured while modelling the domain(s) of interest of the community.

In the 2000s it was the advent of ontology engineering, especially in the semantic Web community, and I addressed the questions of ontology-oriented domain modelling, semantic annotation and semantic search of digital resources mainly in three domains: in e-Education where e-learning environments can be viewed as special cases of organisational memories in a given domain, in the building industry to support the management of the ever growing mass of regulatory documents, and in Life Science to support the capitalization of expert knowledge and scientific heritage promotion and its analysis and exploitation. More recently, I broadened the latter focus to cultural heritage. Nowadays KRR and semantic Web based approaches for managing digital information resources are well known and quite widely adopted or targeted, but some years ago this was a real challenge. During the last 16 years my research activity on this topic

evolved from exploring and demonstrating the feasibility of ontology-oriented and semantic Web based modelling and management of digital resources for various use cases to multidisciplinary projects or industrial partnership aiming to produce effective ontological resources and setup ontology-oriented preservation and exploitation of resources.

For the sake of simplicity, and in a historical perspective, throughout this document, I will (over)use the term *ontology* in its broadest sense, while the general term *vocabulary* would be more correct. When I started working on ontology-oriented modelling 16 years ago, ontology engineering was an emerging scientific field and the notion of ontology itself was still to be properly characterized and refined. Today, we should distinguish between different kinds of vocabularies, with different degrees of semantics, and different degrees of formalization — with controlled vocabularies at the lowest degrees and formal ontologies achieving the highest degrees, and blurred lines between intermediate kinds. In short, a controlled vocabulary is a list of terms gathered for some (modelling) purpose; in a taxonomy these terms are hierarchically organized; in a thesaurus a conceptual level is introduced above terms, with several terms possibly labelling the same concept, and associative relationships between concepts are introduced in addition to hierarchical relationships to build a semantic network of concepts; in an ontology, an instance level is introduced, concepts are formalized into classes of instances and relations between instances, and hierarchical relationships are formalized into subsumption relations between classes and between properties; in a lightweight ontology, classes and properties are *declared*, along with their relationships; in a heavy-weight ontology, classes and properties may be *defined* and subsumption relations can be computed by reasoning on these definitions.

In Artificial Intelligence, a classical partition holds between on one hand *declarative* knowledge, i.e. "passive" knowledge consisting in both factual statements describing the world and ontological statements describing the world model, and on the other hand *procedural* knowledge enabling to perform tasks, to solve problems. However, reasoning on formal representations of digital information resources requires procedural knowledge and raises the question of managing the implicit procedural knowledge of epistemic communities. The birth of a specific International Web Rule Symposium (RuleML) in 2007 shows the evolution of the AI community to insist on the importance of rules, *par excellence* procedural knowledge, as an *object* of study. The publication of W3C standards for knowledge representation and reasoning on the Web followed the same course, starting with languages dedicated to the representation of declarative knowledge — RDF, RDFS, OWL and SPARQL 1.0 query language —, and continuing with languages the formalization of processes — the SPARQL 1.1 recommandation suite, SWRL, SPIN, SHACL. I early addressed this research question of managing procedural knowledge by considering procedural knowledge as a special kind of digital resources which can be semantically annotated to manage them throughout their life-cycle. More specifically I dealt with the capitalisation and management of queries, constraints, requirements, design patterns, legal rules, inference rules, within various epistemic communities, mainly in the above cited application domains.

To sum up, I early addressed the research question of *How can we represent and reason on digital resources?*. I contributed to the adoption of an ontology-oriented approach of domain modelling, enabling to semantically an-

notate digital resources and then reason on their formal representation. My perspective has broadened to consider the modelling of the know-how knowledge of community members dealing with digital information resources: I addressed the research question of *How can we model and capitalize the procedural knowledge of epistemic communities?*. I contributed to the adoption of an integrated ontology-oriented framework to capitalize and manage this knowledge as a special kind of digital resources.

This chapter is organized as follows: Section 1.1 presents my work on ontology-oriented modelling and management of digital resources in the domain of e-Education. Section 1.2 presents my work on ontology-oriented modelling and management of digital resources in the Building Industry. Section 1.3 presents my work on ontology-oriented modelling and management of digital resources in Cultural Heritage projects.

The works synthesized in this chapter have been published in the proceedings of several national French conferences and workshops: Journée *Web sémantique pour le e-Learning* (PFIA 2005) [54][46], Journées *Extraction et Gestion des Connaissances* (EGC 2008) [57], Journées Francophones d'*Ingénierie des Connaissances* (IC 2008, IC 2009, IC 2012, IC 2014) [99][37][6][85], atelier *Visualisation d'information* (IHM 2014) [53], in the proceedings of several international conferences and workshops: *Int. Conf. on Advanced Learning Technologies* (ICALT 2006) [44], *World Conf. on Educational Multimedia, Hypermedia & Telecommunications* (ED-MEDIA 2006) [43], *Int. Conf. CIB W78 Information Technology for Construction* (CIB 2007, CIB 2011) [100][10], *Int. Conf. on Web Information Systems and Technologies* (WEBIST 2009) [102], *European Conf. on Product and Process Modelling* (ECPPM 2010, ECPPM 2012) [9][7] *Int. Workshop on Resource Discovery* (RED 2010) [79], *Int. Conf. on Web Reasoning and Rule Systems* (RR 2011) [8], *ECAI 2012 Workshop Artificial Intelligence meets the Web of Data* (AImWD 2012) [84], *ESWC Int. Workshop Semantic Web for Scientific Heritage* (SW4SH 2015, SW4SH 2016) [92][20][56], in one French journal: *Archivum Latinitatis Medii Aevi* [80], and in three international journals: *Int. Journal of Web-Based Learning and Teaching Technologies* [96], *Int. Journal of Knowledge Engineering and Soft Data Paradigms* [103], *Future Internet* [11].

1.1 Modelling and Managing Digital Resources in e-Education

1.1.1 Ontology-oriented Learning Content Management System (LCMS)

In the framework of Sylvain Dehors's PhD thesis [42], I early addressed the problem of modelling and managing learning resources to support the creation and organisation of learning content. The scenario we considered was the reuse of course material available on the Web, to be adapted using external ontological knowledge and used for learning in a classroom context [54]. To answer the well-known problem of students "getting lost in the hyperspace" of an online course and loosing focus and motivation, we defined the Question Based Learning Strategy, the rationale behind it being that students are motivated by practical questions they have to answer, and we implemented this strategy in

the Question Based Learning System (QBLS) intended to provide learners with a *conceptual navigation* among digital learning resources, guided by a network of concepts. This involved the development of domain ontologies and pedagogical ontologies, and associated semantic annotations of the pedagogical resources selected by a teacher (or a teaching team) [46] [44].

Figure 1.1 presents the overall approach adopted in the conception of the QBLS system. It was similar to the approach adopted in other existing e-learning Systems; its originality lied in the use of semantic Web models and technologies to improve both the author's and learner's experience.

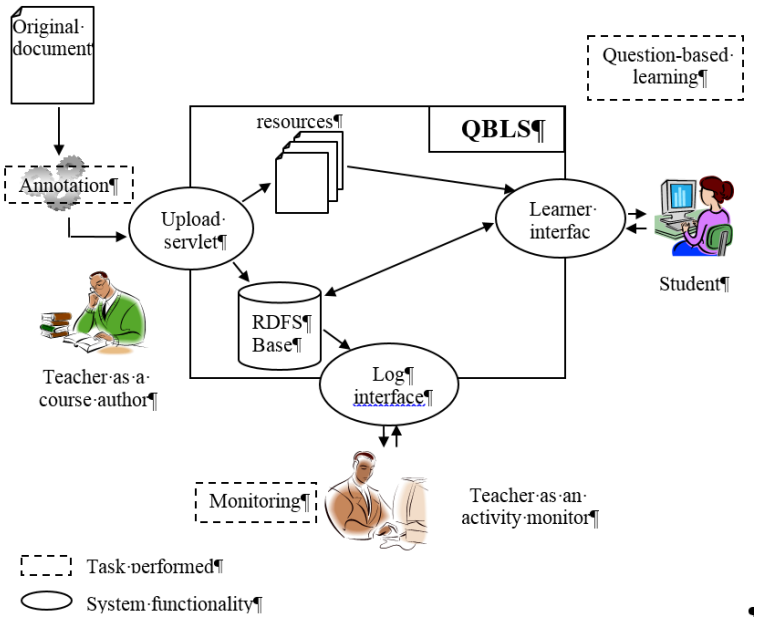

Figure 1.1: Overall QBLS approach for creating and exploiting learning resources from available course material

Ontology-oriented Modelling of Pedagogical Resources

Our approach to reuse pedagogical material relied on two complementary annotation processes: a first one based on ontologies modelling the learning domain covered by the material, and a second one based on a pedagogical ontology modelling question-based learning. We developed a SKOS thesaurus for an introductory course on Java programming, partially reusing ontologies available on the Web, another thesaurus for a course on signal analysis, and an OWL lite ontology to formalize the *pedagogical* model. The annotation process relating to the domain model was automated, based on the detection of concept labels occurring in the document (this is what is now called Named Entity Recognition (NER)). It was a basic approach, without advanced Natural Language Processing.

For the annotation process relating to the pedagogical model, the course author was assisted by exploiting the layout of the documents: assuming that

two elements in a document playing the same role are similarly laid out, we conducted a kind of reverse engineering process on the intention of the document's author to automatically extract semantic annotations from the text layout. The document model was mapped with the pedagogical model by defining text styles in text editors, making the link between semiotic markers and semantics [43]. Figure 1.2 shows the annotation of learning resources with OpenOffice Writer and its style hierarchy corresponding to the concept hierarchy in the ontology. Internal styles are configured in the document file itself using a specific XML representation. The styles for a specific model are automatically configured by transforming the ontology (in the RDF/XML syntax) with an XSL stylesheet to produce the required XML data to be integrated in the XML representation of the OpenOffice document. The questions intended to initiate the conceptual navigation in the course materials were available on a wiki and annotated according to a similar process.

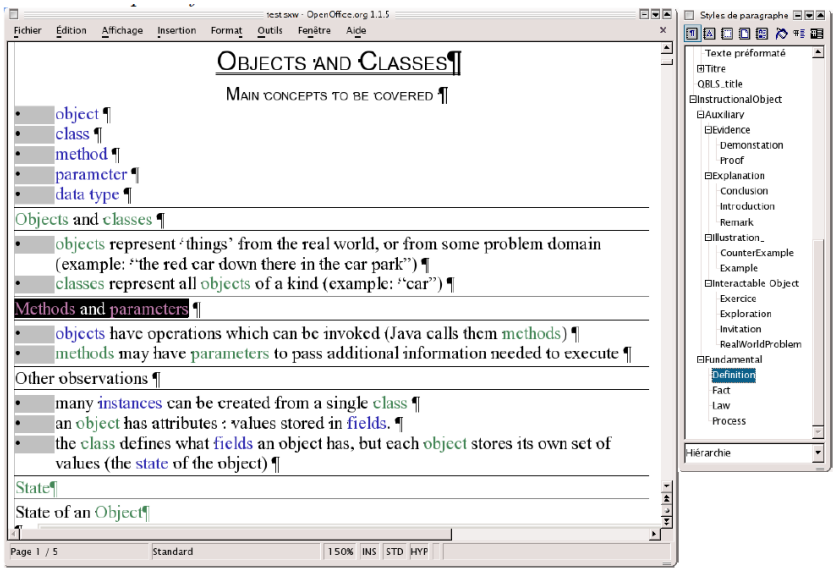

Figure 1.2: Annotation of learning resources with OpenOffice Writer and its style hierarchy

Sylvain Dehors also developed a dedicated annotation editor to enable updating or enriching annotated pedagogical resources without going back to their OpenOffice original format. The editor was built on the following principle: it takes an RDF/XML file as input, and processes it through an XSL stylesheet to generate an HTML page where every RDF element (node, relation) is a dynamic link. By clicking on an element, a form is dynamically generated, taking into account the ontology and the context in which the element appears, and allowing the user to edit the selected element. While editing an annotation, the user is guided by exploiting property domains and ranges declared in the ontology as constraints. Figure 1.3 is a screenshot of QBLS annotation tool.

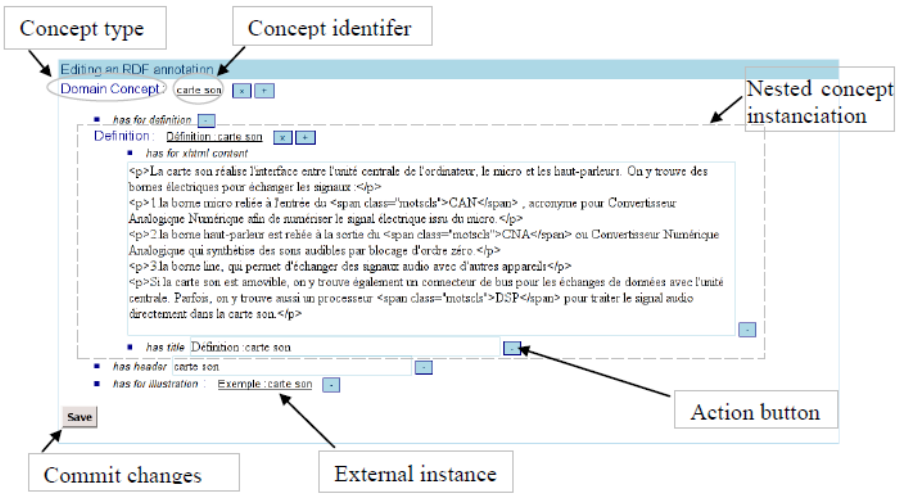

Figure 1.3: QBLS annotation tool

Ontology-oriented Navigation through Pedagogical Resources

We then developed an e-learning environment based on semantic Web models and technologies, where the navigation over learning resources relies on the semantic querying of the RDF annotations of resources with the SPARQL language. We capitalised SPARQL query patterns which where dynamically instanciated throughout the user navigation in order to dynamically compute the next learning paths, depending on the relationships of the pedagogical resource currently visited and other resources in the domain and the pedagogical ontologies, and on the user context [45]. Contrary to the use cases addressed in the Building Industry (see Section 1.2.3), these query patterns were not meant to be shared and reused outside the system, nor presented to human end-users as valuable expert knowledge.

Figure 1.4 is a screenshot of the QBLS interface developed for navigating through a Java course. The user accesses it from the wiki of lab questions, when clicking on the occurrence of a domain concept he wants to explore. The window is divided in four zones: (1) a banner at the top indicating the current chapter; (2) a central area displaying the resources relative to the chosen concept. On this example, four resources are presented, relative to the concept of `Object`. Resources are organized in tabs, each tab header corresponding to a pedagogical concept. On this example, the visible resource is of type `Definition`; (3) the right part contains a list of previously visited concepts; (4) at the bottom of the screen, a `Back` button allows to go back in the navigation history. Hyperlinks within the resources enable to access resources related to other related domain concepts. Let us note that the dynamic composition of such pages integrates heterogeneous knowledge: about the resources (titles, types), the user history (list on the right), and the context indicated by the current chapter (at the top). This was a first step towards adaptive learning, quite new in the 200X's. For instance, the QBLS interface proposes a feature of "link hiding" consisting in hiding links that are not relevant in the current context (the chapter being studied): for instance, for introductory chapters, it prevents the learner to navigate

from one concept described in the chapter to another one described in another chapter.

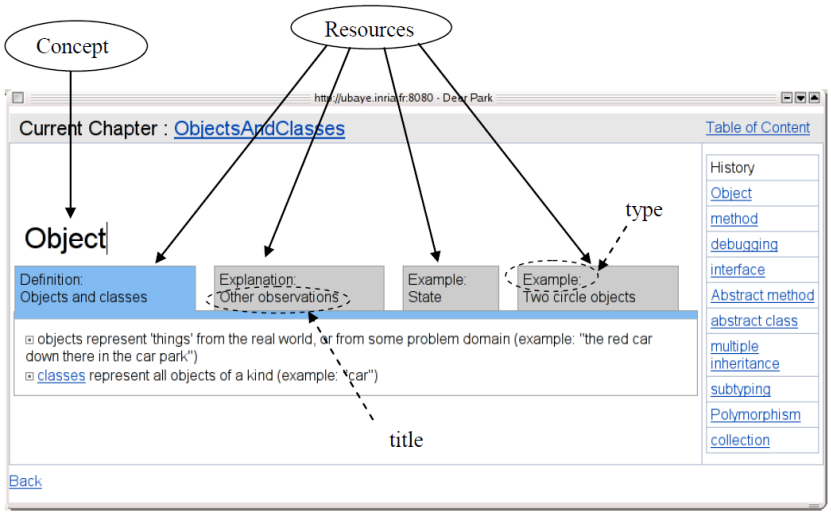

Figure 1.4: QBLS interface for navigating a Java course

Constructing and Visualizing Graphs of Concepts

We constructed a graph of concepts representing a course considering both ontological relations between domain concepts and relations that can be inferred between concepts by considering `rdfs:seeAlso` relations between concept instances (the pedagogical resources) and other domain concepts, e.g. between concept `Object` and concept `Class` in Figure 1.4. Figure 1.5 presents the graph of concepts generated for the Java course. We adopted a complete Web approach for the processing of the RDF graph: it was transformed with an XSL stylesheet to generate DOT data given as input to Graphviz to generate an SVG graph. Such resulting graph may serve as a visual diagnosis tool for the course author to check the course coherence, to spot isolated concepts, loops, etc. In the QBLS analyzer interface for teachers, the instanciation of a predefined SPARQL query template enabled to select a concept and visualize the only resources accessible from this concept in the conceptual graph.

We also experimented the use of this conceptual graph as a base structure to analyse the actual use of the course content and help the teacher monitor the classroom (see Chapter 2).

1.1.2 Ontology-oriented Adaptive Learning Environment

In the framework of Amel Yessad's PhD thesis [95], I continued addressing the problem of modelling and managing learning resources to support the organisation of learning content. When compared to our research work on the design of the QBLS system, (1) we did not consider the creation of learning resources as a primary goal and we considered instead the reuse of resources available

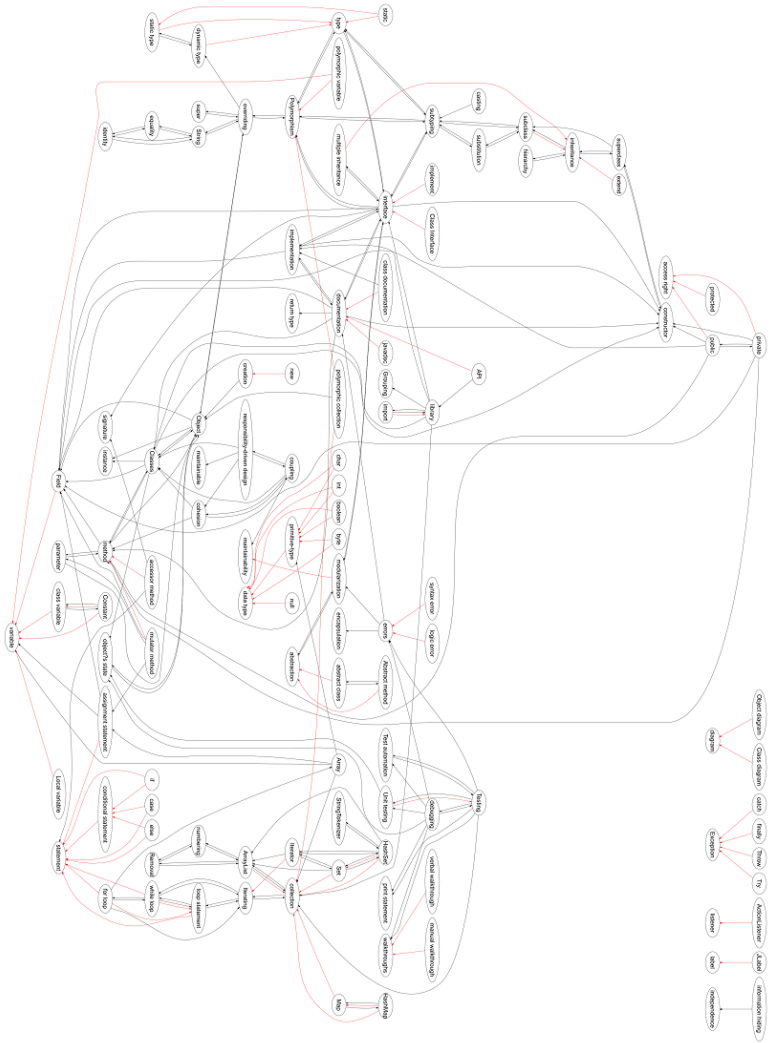

Figure 1.5: Graph of concepts of the Java course

on the Web, and more specifically resources retrieved from the ARIADNE[1] repository, already semantically annotated with the Learning Object Metadata (LOM) IEEE standard[2], to be adapted using additional ontological knowledge. Moreover, (2) we considered a pedagogical model based on the constructivism theory where the learner is at the very center of the learning process, active and in charge of the construction of his/her knowledge. As a result, (3) we conceived the OrPAF learning organizer (*Organisateur de Parcours Adaptatifs de Formation*), with the aim of providing the learner with an *adaptive conceptual map* representing his/her personal learning space to achieve his/her learning goal. This is further discussed in Chapter 2.

When compared to QBLS model, OrPAF model goes a step further in generalizing the ontology-oriented modelling of learning systems. We developed a meta-model, formalized as an ontology, which represents the generic concepts of e-Learning, irrespective of any targeted training [96]. Its top class is specialized into four classes: the class of learning topics, the class of learning pedagogies, the class of learning actors and the class of learning resources. Generic properties are defined to describe instances of these classes. To model a specific learning system, for a specific learning domain, a specific learning context (initial or continuing training) and its associated learning strategy (related to constructivist learning), and a specific learner, this meta-model is instantiated with an ontology-oriented model of the learning domain, an ontology-oriented model of the learning strategy, an ontology-oriented model of the learner, and annotated learning resources. Figure 1.6 presents an excerpt of our modelling for an OrPAF instance in Algebra, with specialisation relations and prerequisite relations between domain concepts, and a pedagogical model declaring various types of activities and relations between them enabling to sequence them.

For the development of the ontologies, we used the ontology editor of the at-the-time KAON ontology management infrastructure[3]. For the construction of the domain model, we adopted a semi-automated approach based on text mining (Named Entity Extraction on a corpus of chosen texts); we used the TextToOnto ontology learning environment included in KAON.

Adaptive conceptual maps are subgraphs of the conceptual map consisting of the graph of domain concepts linked with specialization, prerequisite and aggregate relations. Several learning paths may be possible between two domain concepts in this graph and the learner is free to chose his/her path when navigating on the graph to achieve his/her learning goal.

Distant learning resources available on the Web are selected by querying the ARIADNE repository; they are stored as local resources and manually annotated by the teachers. Just like in QBLS, the learning resources in OrPAF are annotated by using the domain model and the pedagogical domain, but also the learner profile: learning resources are indexed by domain concepts (e.g. Symmetry), pedagogical activities (e.g. Formal definition) and learner preferences (e.g. language, format, author).

In OrPAF, the presentation of learning resources to the learner is secondary to the presentation of the domain concepts into a conceptual map adapted to the learner profile and context. This can be viewed as a continuation of our experiment to exploit the graph representation of the course content. In

[1] `http://www.ariadne-eu.org/`

[2] `https://standards.ieee.org/findstds/standard/1484.12.1-2002.html`

[3] `http://kaon2.semanticweb.org/`

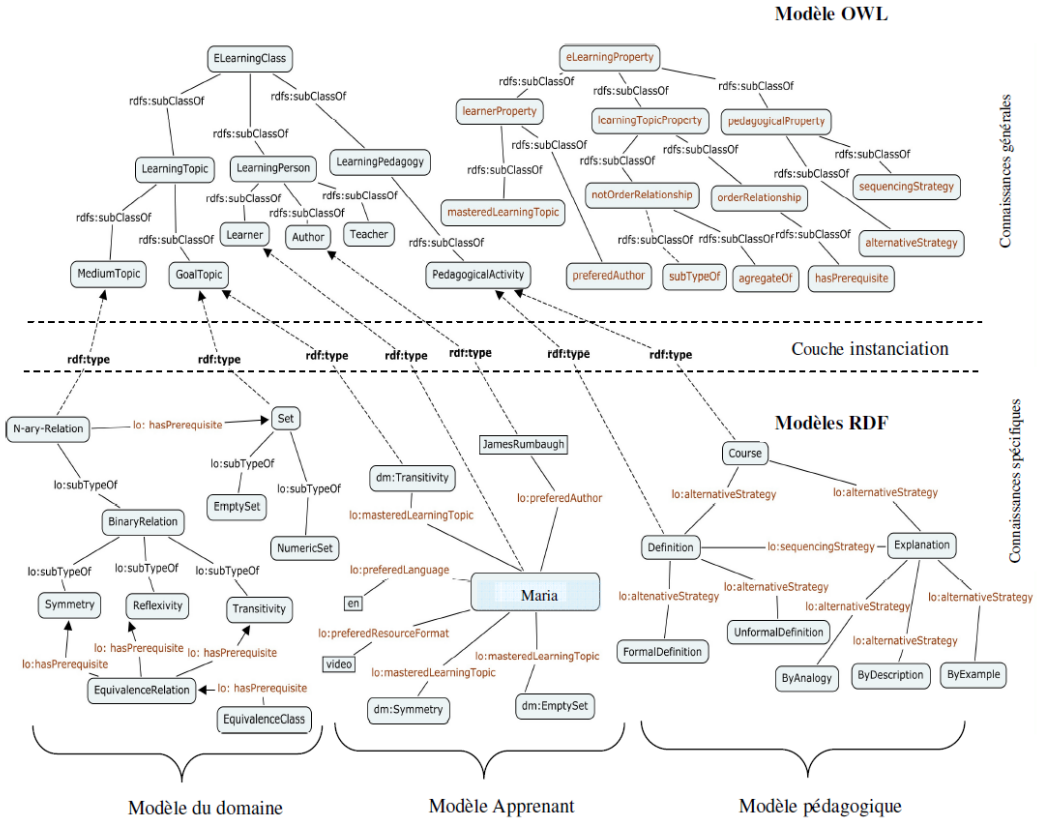

Figure 1.6: The OrPAF meta-model and models

QBLS, this graph representation was intended for the teacher to support his/her analysis of the learning activity; in OrPAF, it is intended for the learner to support his/her learning activity. The construction of this adaptive conceptual map and the selection of learning resources indexed to the learning domain concepts are described in chapter 2.

1.2 Modelling and Managing Digital Resources in the Building Industry

1.2.1 Ontology-oriented Modelling of Construction Projects and Construction Norms

In the framework of Anastasiya Yurchyshyna's PhD thesis, we addressed the problem of supporting the conformance checking of construction projects against construction norms, and we developed an ontology capturing key concepts of the building industry domain [98]. It was based on the Industry Foundation Classes (IFC) standard, an object-oriented data model in the architecture, engineering and construction (AEC) industry and we limited it to the only concepts occuring in technical construction norms [100] [101].

We used this conformance-checking oriented ontology to semantically annotate documents describing construction projects and to formalize the constraints expressed in the construction norms in order to enable ontology-oriented reasoning on the formal representation of these construction projects. As a result, conformance checking was supported by an automatic process consisting in matching the formal representations of construction projects and normative constraints to retrieve conform projects or non conform projects and which parts of them violate a given norm [57].

Throughout this knowledge management process, we used semantic Web models and technologies. The ontology was formalized in OWL-Lite, construction projects were annotated in RDF, construction norms were represented in SPARQL and conformance checking consisted in querying these RDF annotations with these SPARQL queries, and interpreting in terms of conformance checking in construction. Nowadays, ontology-oriented modelling and semantic Web models are widely spread in many domains. Back to 2007, when we started applying this approach to model the domain knowledge and semantically annotate digital resources in the building industry, this was a scientific contribution. Our work and publications in international reference conferences in the building industry (CIB W78[4] and ECPPM[5]) contributed to the adoption of ontology-oriented approaches and semantic Web standards for building information modelling (BIM) in CSTB and more generally in the building industry. Moreover, our work was a pioneer attempt to propose an approach to validate RDF data, well before the creation of the RDF Data Shapes working group in 2014.

Our approach to construct the ontology was following the general reference methodology proposed by Uschold and Gruninger [93] identifying the key concepts of the ontology guided by its purpose — conformance checking —, reusing

[4] http://cibw78.org
[5] http://www.ecppm.org

existing resources — the IFC model —, and formalizing it — in OWL-lite. Basically, it consisted in the following steps. We first selected the parts of the IFC model related to conformance checking. Then we considered the XML representation of the IFC model (IFC-XML), and its model represented in XML Schema, and we wrote an XSL stylesheet to automatically construct an OWL-Lite ontology from the XSD schema: XSD types represent IFC classes and are transformed into OWL classes; XSD elements occurring in the definition of a complex type represent IFC class attributes and are transformed into OWL properties; the definition of XSD types are transformed into OWL class definitions. For instance Listing 1.1 shows the definition of the IFC class `Door` in XSD and Listing 1.2 its transformation in OWL/XML.

```
<xs:complexType name="IfcDoor">
  <xs:complexContent>
    <xs:extension base="ifc:IfcBuildingElement">
      <xs:sequence>
        <xs:element name="OverallHeight" minOccurs="0"
            type="ifc:IfcPositiveLengthMeasure" nillable="true">
        </xs:element>
        <xs:element name="OverallWidth" minOccurs="0"
            type="ifc:IfcPositiveLengthMeasure" nillable="true">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Listing 1.1: Definition of the IFC class `Door` in XSD

```
<owl:ObjectProperty rdf:about="#overallHeight">
  <rdfs:domain rdf:resource="#IfcDoor"/>
  <rdfs:range rdf:resource="&IFC2x3;IfcPositiveLengthMeasure"/>
</owl:ObjectProperty>
<owl:Class rdf:about="#IfcDoor">
  <rdfs:subClassOf rdf:resource="#IfcBuildingElement"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="&IFC2x3;IfcPositiveLengthMeasure"/>
      <owl:onProperty rdf:resource="overallHeight"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="&IFC2x3;IfcPositiveLengthMeasure"/>
      <owl:onProperty rdf:resource=="overallWidth"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Listing 1.2: Definition of the IFC class `Door` in OWL

These class definitions were manually refined and enriched with labels and comments. In addition, the special purpose of constraint checking required to model some knowledge out of the scope of the BIM, and consequently, not expressed

in the IFC model. To explicit this expert knowledge, we interviewed construction experts and expressed new non IFC concepts in terms of the IFC model. For instance, in order to model and check accessibility rules, we introduced class `SpecialisedSpace` (and its subclasses: `PublicSpace`, `InteriorSpace`, `DoorSpace`, etc.) as a subclass of `IfcSpace`, `IfcVirtualElement` and `IfcZone`, which inherits their definitions (restriction of property `floorCovering` for class `IfcSpace` and restriction of property `hasOpenings` for `IfcVirtualElement`).

Our approach to acquire RDF descriptions of construction projects was also based on the automatic transformation of their IFC-XML representation, with XSL stylesheets. This initial representation was then enriched by applying in forward chaining the rules defining the non IFC classes of the ontology (more specific types were inferred for some resources). Finally, it was simplified by filtering the RDF triples relevant for conformity checking with the ontology (triples with properties or classes which were not in the ontology were removed).

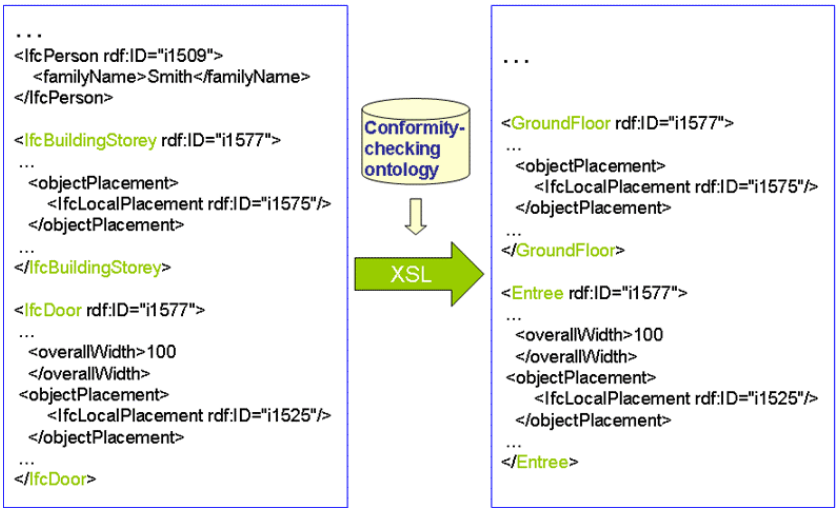

Figure 1.7: Refining the RDF description of construction projects

1.2.2 Ontology-oriented Modelling of Technical Documents and Technical Regulation

From 2010, in the framework of Khalil Bouzidi's PhD thesis, we addressed the problem of modelling technical documents and technical standards in the building industry to support the writing of technical documents by industrials and their technical assessment by CSTB experts with respect to technical standards [5]. A Technical Assessment (in French: Avis Technique or ATec) is a document containing technical information on the usability of a product, material, component or element of construction, which has an innovative character. CSTB has the mastership and a wide experience in technical assessment in the Building Industry. A technical assessment is established by CSTB at the request of an industrial. In 2010, to help industrials writing their technical documents, CSTB provided them a preformatted Word file containing chapters, text and instructions on how to fill it out. The completed document was supposed to

describe with the right accuracy the process, product or material candidate for a Technical Assessment. This technical document was then studied by a specialized group, responsible for delivering the technical assessment.

Similarly to the approach proposed in Anastasiya Yurchyshyna's PhD thesis, her we developped a conformance-checking oriented ontology to semantically annotate technical documents: OntoDT aimed at capturing the main concepts of technical guides published by CSTB and offering industrials some regulatory complements enabling an easier reading of technical rules [9], e.g. "Les couvertures en tuiles"[6]. At that time, we could rely on a SKOS thesaurus newly developed by CSTB within the REEF[7]. It helped us build the OntoDT ontology from REEF terms and their hierarchical organisation. We started by manually eliciting domain terms from (1) the general template provided by CSTB to industrials to write a technical document and (2) the interviews conducted with experts from the CSTB group specialised in photovoltaics. The alignment of the elicited terms with those present in the REEF thesaurus helped us structure our lightweight ontology. Then we enriched it with *defined* classes, based on structural and dimensional criteria extracted from the technical document template and the technical guide. For instance, we defined the module "Polymer glass" as a special kind of photovoltaic module, having several components, among which a module "Photovoltaic cell" and another "Photovoltaic film" (Figure 1.8). Formally, `odt:VerrePolymere` is a subclass of`odt:ModulePhotovoltaique` and of the class defined as the intersection of property restriction, each gathering individuals sharing a given component.

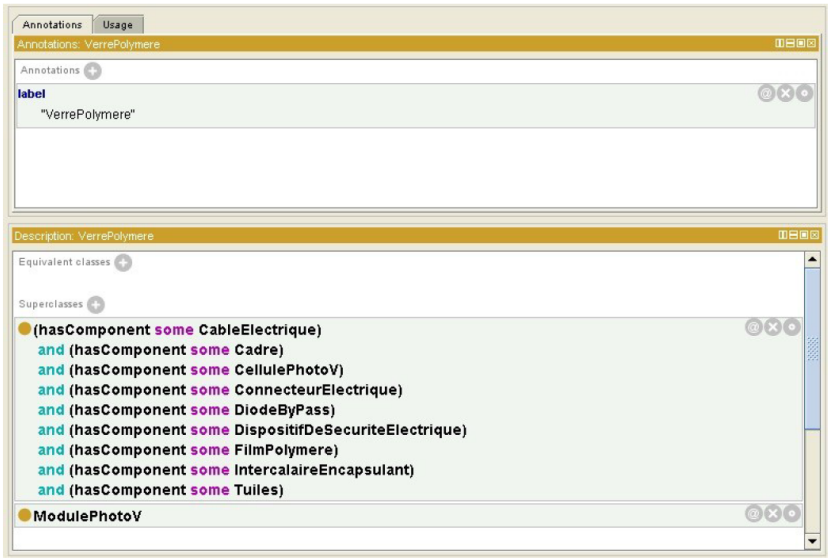

Figure 1.8: Definition of class *Polymer Glass* in the Protégé ontology editor

Similarly to the approach proposed in Anastasiya Yurchyshyna's PhD thesis,

[6] `https://boutique.cstb.fr/guide-pratique/88-les-couvertures-en-tuiles-9782868916693.html`

[7] Recueil des Eléments utiles à l'Etablissement et l'exécution des projets et marchés de bâtiments en France, `https://boutique.cstb.fr/4-gamme-reef`

we used the OntoDT ontology both to semantically annotate in RDF technical documents and to represent in SPARQL the constraints expressed in the CSTB technical guides, with the aim of validating technical documents against technical regulations [10]. We developed an automatic process to support the technical assessment of technical documents by CSTB experts, consisting in automatically matching the formal representations of technical documents and normative constraints. This will be further described in the following subsection.

Upstream, we also used OntoDT to support the writing of technical documents, by dynamically generating and combining the Web forms associated to the elementary components of the product to be assessed. This relies on the recursive call and instanciation of the SPARQL query pattern shown in Listing 1.3, where variable `?class` must be each time instanciated with an OntoDT class name to query OntoDT.

```
SELECT ?component WHERE {
?class rdfs:subClassOf ?y
?y owl:intersectionOf ?z
?z rdf:rest*/rdf:first ?f
?f owl:onProperty odt:hasComponent
?f owl:someValuesFrom ?component }
```

Listing 1.3: SPARQL query pattern to automatically generate Web forms to describe the products to be assessed

Figure 1.9 presents the overall process of assisting the writing of technical documents. Its output is (1) a technical document in Natural Language, built by concatenating the filled forms, and an RDF annotation of this document, also built from the filled forms, the classes associated to the forms, and the properties associated to the form fields.

1.2.3 Ontology-oriented Modelling of Expert Knowledge for Conformance Checking

The use cases we tackled in the building industry led us to the question of capitalising and managing a special kind of knowledge, namely *procedural* conformance-related knowledge, first for semi-automatic checking of construction projects and then for supporting the writing and assessment of technical documents.

Modelling Conformance Requirements

For the semi-automatic checking of the conformance of construction projects against a set of applicable standards, we (manually) extracted conformance requirements from regulatory texts and we formalized them as SPARQL queries enabling to query the RDF representation of construction projects. We considered these elementary conformance requirements as a special kind of knowledge which should be capitalized, shared and reused in the building industry community [103] [102] [99]. To achieve this goal, we developed a special vocabulary to annotate conformance requirements so that they can be retrieved with meta-queries. These annotations comprised characteristics of the regulation text from which the query was extracted (thematic, type of regulation text, level of application, etc.), characteristics of the specific part of the regulation text which is represented by the conformity query (article, paragraph),

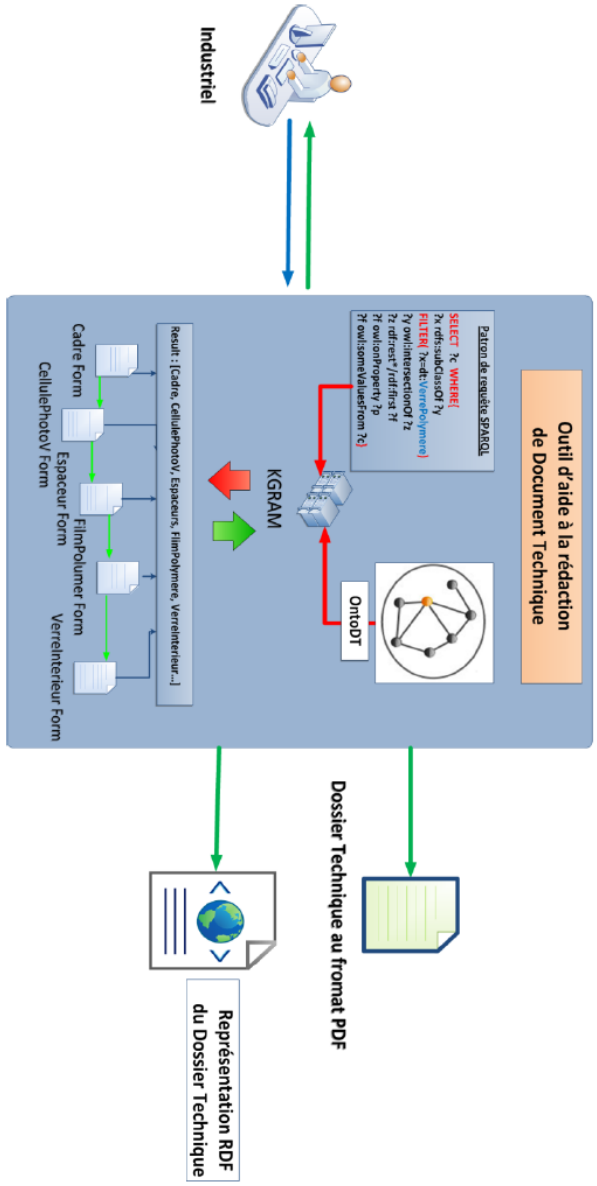

Figure 1.9: Assisting the writing of technical documents

the content of the requirement itself (domain of application, informal reformulation of the requirement with IFC terms), the SPARQL representation of the requirement, the context of application of a requirement (e.g. the requirements on the maximal height of stairs handrails depends on the nature of a building, i.e. public administration, school, etc., it varies from 96 cm for adults to 76 cm for kids). Listing 1.4 shows a simplified example of the semantic annotation of a requirement. Let us note that at the time we conducted this work, the SPARQL Inferencing Notation (SPIN[8]) was still to come; today, the literal values corresponding to SPARQL codes would be replaced by the RDF/SPIN representation of the SPARQL queries.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://ontologyQueriesSemanticAnnotations.owl"
    xmlns:ontoCC="http://conformityCheckingOntology.owl#">
  <QueryAnnotation rdf:ID="r00020020q">
    <differentApplicationContext rdf:parseType="Collection">
      <rdf:Description
        rdf:about="http://conformityCheckingOntology.owl#Children"/>
      <rdf:Description
        rdf:about="http://conformityCheckingOntology.owl#Adults"/>
    </differentApplicationContext>
    <queryContent rdf:ID="r00020020q_children">
      <!--A SPARQL query using ontoCC concepts-->
    </queryContent>
    <queryContent rdf:ID="r00020020q_adults">
      <!--Another SPARQL query using ontoCC concepts-->
    </queryContent>
    ...
  </QueryAnnotation>
</rdf:RDF>
```

Listing 1.4: Simplified example of the semantic annotation of a conformance requirement

Then we interviewed CSTB experts and we captured their "know-how" relative to the conformity checking process into SPARQL (meta)queries enabling to classify conformity constraints (e.g. retrieving accessibility requirements extracted from circulars, or requirements relative to lifts in public buildings) and order them (e.g. checking conformity requirements relative to schools before those relative to public building, because if the former are not met, it may be useless to test the later). Finally, based on these classifications and ordering, we scheduled the conformity queries handled in our algorithm for checking the conformity of a whole construction project. Its RDF description is iteratively matched with the list of selected and ordered SPARQL queries and a conformity report is generated gathering the results of the queries.

Similarly, for supporting the writing and assessment of technical documents [11], we (manually) extracted constraints from the technical regulation described in CSTB guides and we formalized them as SPARQL queries enabling to query the RDF representation of technical documents. Here again, we answered the question of managing this procedural knowledge of the building industry community by identifying the extracted regulatory constraints by URIs and semanti-

[8] `https://www.w3.org/Submission/spin-overview/`

cally annotating them so that they can be retrieved with meta-queries. For this modelisation of technical regulation, we improved our approach proposed for the modelisation of construction regulation, by better keeping the link with the original text through a semi-formal representation of it, in the Semantics of Business Vocabulary and Business Rules (SBVR) language, based on a controled vocabulary. SPARQL queries are extracted from these SBVR representations and linked to them, enabling to improve the interaction with the user when reporting on the conformance checking results in Natural Language [8]. Figure 1.10 shows the overall process of extracting SBVR representations of technical rules from texts (here a summary table of slopes and overlapping lengths for flat tiles, in the CSTB guide *"Les couvertures en tuiles"*), and SPARQL representations from SBVR rules.

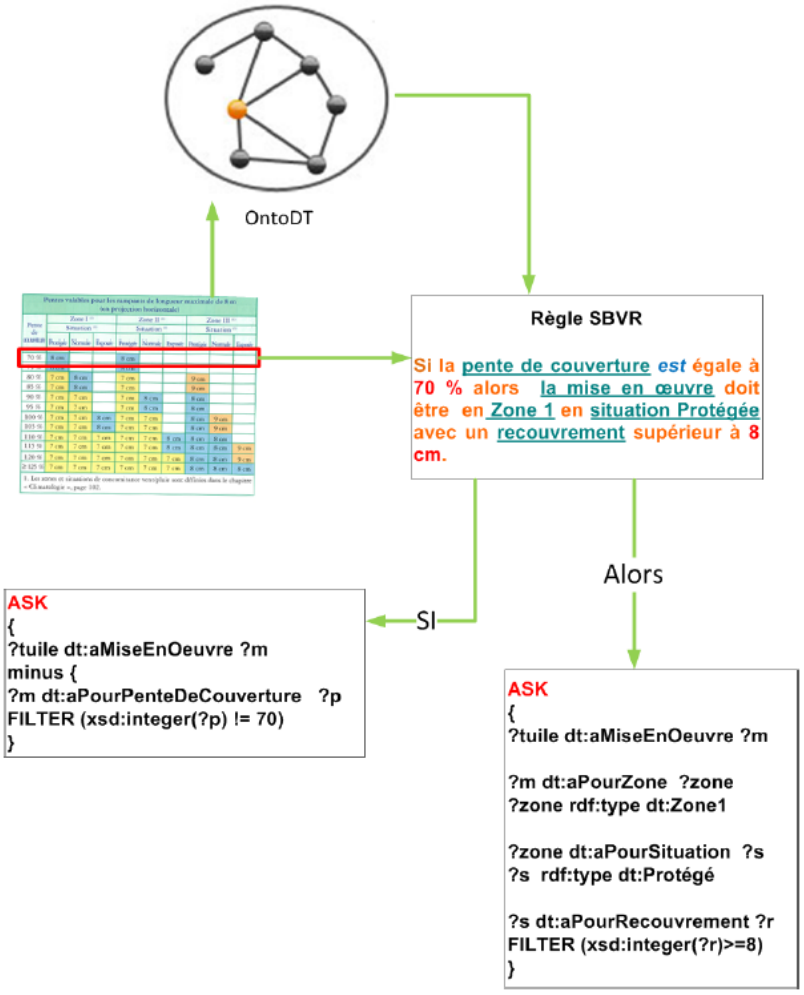

Figure 1.10: Assisting the writing of technical documents

Modelling Conformance Checking Processes

Regarding the capture of the CSTB experts while assessing technical documents, we went a step further than in our work on the conformance checking of construction projects: we proposed a general process model and used it to declaratively represent a process of validating a technical document against regulatory constraints [6] [7] [11].

Relatedly, three years earlier, with my colleague Olivier Corby and Isabelle Mirbel we had proposed an approach to capitalize, share and reuse search processes and guidelines to compose sequences of queries enabling to find comprehensive information [38][37]. Starting from the Map model, an intention driven process modelling formalism, we proposed an ontology based model to represent search processes in RDF and search guidelines associated to search process fragments with rules formalized as SPARQL queries. As a result, the instantiation of search processes is supported by backward chaining on the rule base and matching with the RDF dataset annotating the community resources. But this model had not been implemented and tested.

The model proposed in the context of our collaboration with the CSTB was simpler and operationalized. It relies on a small lightweight vocabulary comprising 4 properties: `body`, `if`, `then` and `else`, and 5 classes: `Load`, `Query`, `Pipeline`, `Pipe` and `Test`. The main class, `Pipeline`, represents a process and class `Pipe` represents the call to a process. Class `Query` represents a SPARQL query. The execution of queries or rules can be conditional: this is represented with class `Test` and properties `if` and `then` which values are processes to be executed. The description of a process can then recursively call other processes (`Pipe`), included the process itself. The abstract syntax of a process is defined by the following grammar:

```
PIPELINE ::= EXP +
EXP ::= Load(Name) | Query(Name) | Test(Query(Name), Exp, Exp) | Pipe(Name)
```

We distinguish between elementary and complex processes. An elementary process is associated to a component occurring in a technical document which is represented in the domain ontology by an *atomic* class. It consists in the compliance checking of the attributes of the component and is represented by the set of necessary SPARQL queries. For instance Listing 1.5 an excerpt of the RDF representation of the elementary process to check the conformity of a tile slope; it comprises the representation of 13 rules:

```
@prefix kg: <http://ns.inria.fr/edelweiss/2010/kgram/> .
@prefix odt: <http://www.cstb.fr/ontologies/odt#> .
@prefix dt: <http://www.cstb.fr/ontologies/data#> .
dt:process1 a kg:Pipeline ;
    odt:composant odt:Pente ;
    kg:body (dt:rule1 ; dt:rule2 ; ... dt:rule13) .
dt:rule1 a kg:Test ;
    kg:if dt:P70 ;
    kg:then dt:Check70 ;
    odt:hasSBVRrule dt:sbvr1 ;
dt:sbvr1 odt:hasValue "Si la mise en oeuvre est dans une Zone 1
    en situation Protégée et un recouvrement supérieur ou égal à
    8 cm ou dans une Zone 2 avec une situation Protégée et un
    recouvrement supérieur ou égal à 8 cm alors la pente de
    couverture est égale à 70 % (Extrait du guide pratique
    "des couvertures en tuiles" en application des DTU 40.2,
    40.211, 40.22, 40,23 et DTU 40.24, 40.241, 40.25)" .
dt:rule2 a kg:Test ;
...
```

Listing 1.5: RDF representation of the elementary process to check the conformity of a tile slope

A complex process is associated to a component *defined* in the ontology as a combination of sub-components. It is recursively defined as a sequence of elementary processes relative to the sub-components. For instance, Listing 1.6 shows the RDF representation of the complex process to check the module "Polymer Glass":

```
@prefix kg: <http://ns.inria.fr/edelweiss/2010/kgram/> .
@prefix dt: <http://www.cstb.fr/ontologies/data#> .
<process> a kg:Pipeline ;
    kg:body ( dt:Pente; dt:Cadre; dt:CellulePV ) .
dt:Cadre a kg:Pipe .
dt:CellulePV a kg:Pipe .
dt:Pente a kg:Pipe .
```

Listing 1.6: RDF representation of the complex process to check the module "Polymer Glass"

Our model enables to build an RDF description of the sequence of constraint verifications which must be performed for a given technical document and we developed a process engine able to automatically build it and launch it. Based on the Corese/KGRAM semantic engine, it analyses the RDF representation of a process, recursively calls sub-process descriptions and dynamically constructs and executes the sequence of SPARQL queries or rules implementing elementary sub-processes (Figure 1.11). It thus enables to supervise, coordinate and sequentially execute a set of queries and rules. The process management relies on a set of predefined SPARQL queries which enables to list all the components of a process and their types.

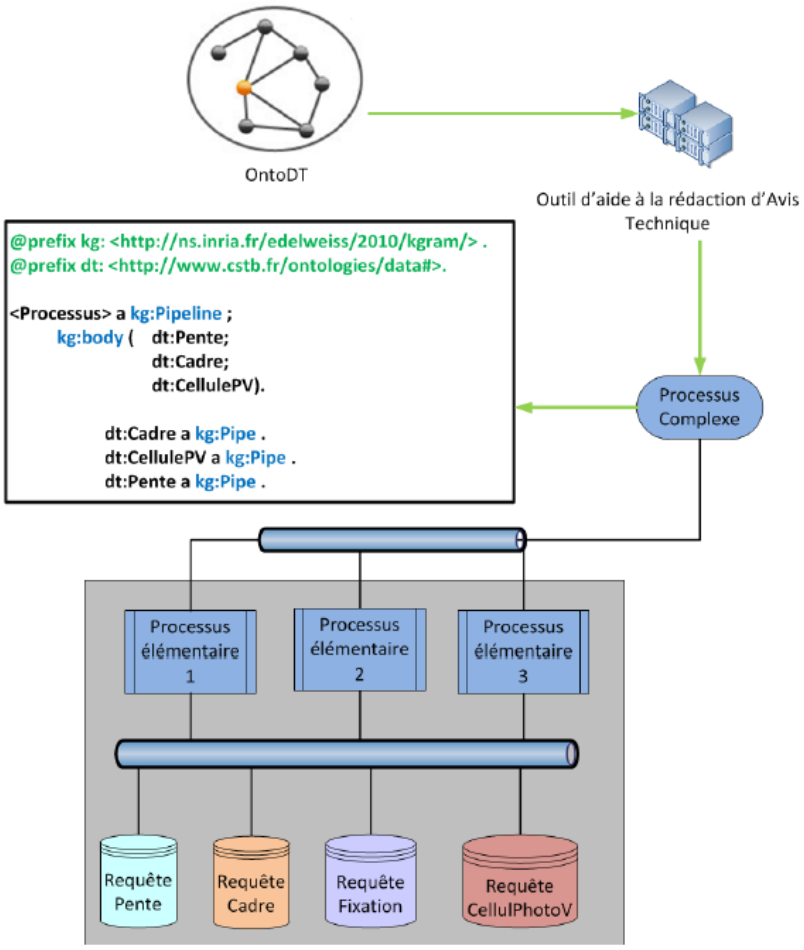

Figure 1.11: Processing of the RDF description of a complex process

1.3 Modelling and Managing Digital Resources for Cultural Heritage

1.3.1 Ontology-oriented Scientific and Natural Heritage

In 2013, I participated to the set up of the international and multidisciplinary research network Zoomathia[9], which aims at studying the transmission of zoological knowledge from Antiquity to Middle Ages through various resources, and considers especially textual information, including compilation literature such as encyclopaedias. In this context, I initiated in Wimmics a research activity aiming at (i) extracting pertinent knowledge from antique and mediaeval texts using Natural Language Processing (NLP) methods, (ii) semantically enriching semi-structured zoological data and publishing it as an RDF dataset and its vocabulary, linked to other relevant LOD set, and (iii) reasoning on this linked RDF data to support researchers in Humanities (epistemologists, historians and philologists) in the analysis of these ancient texts, putting the overall project de facto in Digital Humanities.

As a start, in the framework of Molka Tounsi's Master thesis, which I co-supervised with my colleague Elena Cabrio, we conducted a preliminary work on the zoological mediaeval encyclopaedia *Hortus Sanitatis* and a classical Latin book on fishes (Pliny, *Historia Naturalis*, book IX), which is a major, though indirect, source of *Hortus Sanitatis*. This work aimed at demonstrating to researchers in Humanities the overall collaborative knowledge engineering process that could be conducted to support the analysis of ancient text and the study of the evolution of knowledge through times, from one author to another, from one text to another, especially when considering zoological knowledge and the human-animal relation described in compilation literature: the final aim is to support an accurate evaluation and interpretation of the development of the zoological discourse through two millennia. This work combined state-of-the-art NLP techniques to extract zoonyms — the common names of animals — and animal properties from texts (namely the writing of lexico-syntactic patterns, using WordNet and BabelNet terminological sources), and knowledge engineering and semantic Web methods to build a linked RDF dataset of zoological annotations of these scientific texts, and to exploit this dataset with SPARQL queries to support the analysis of the Ancient zoological knowledge compiled in the encyclopaedia [92]. We also demonstrated the possibilities of supporting a visual analysis of this knowledge by generating graphs showing the existing relations between authors, books, zoonyms, animal properties, and showing the relative importance of zoonyms or animal properties in a given text or author [53]. For instance, Figure 1.12 presents an RDF graph capturing the relative importance of zoonyms in the *Hortus Sanitatis* and their location in it. At a glance, it shows that dolphins, whales and eels occupy a predominant place in this text, far ahead of other animals.

Perhaps most importantly, this work showed within the research network the prime importance of collaborating to develop reference vocabularies to annotate the available heterogeneous resources, beginning with textual resources. As a result, I started two collaborative ontology engineering works, one with classicists to build a thesaurus of zoological knowledge for Zoomathia, and another work

[9] `http://www.cepam.cnrs.fr/zoomathia`

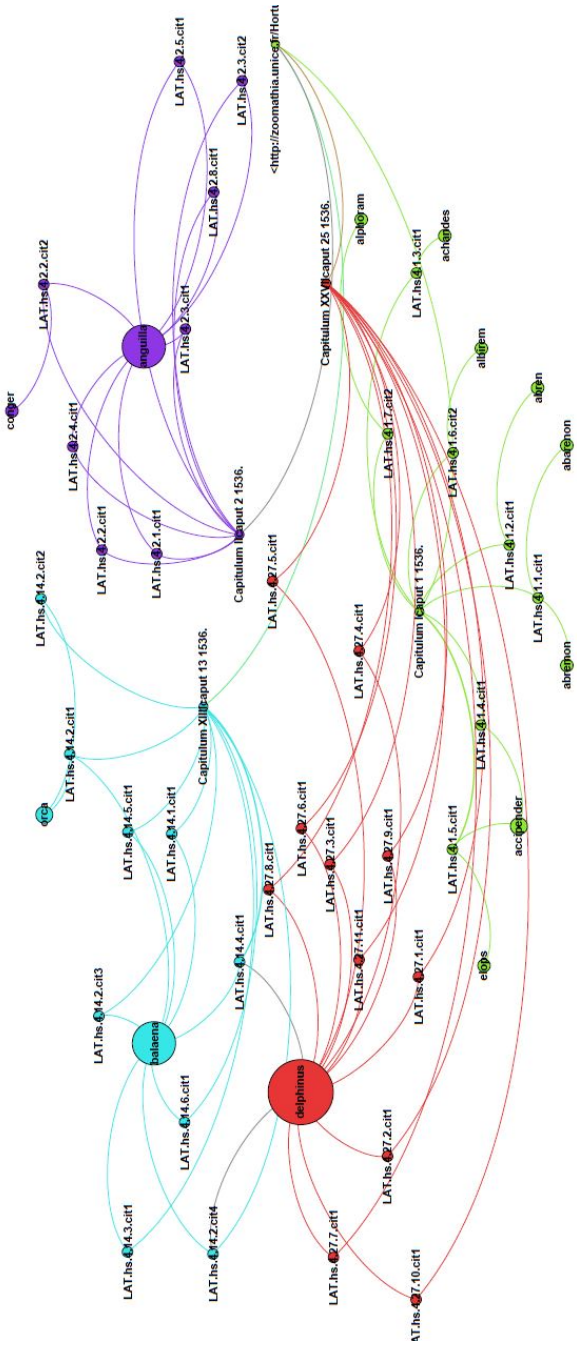

Figure 1.12: Relative importance of zoonyms in *Hortus Sanitatis*

with biologists and archaeologists to publish the TAXREF reference thesaurus for Conservation Biology as a SKOS vocabulary (zoonyms in the Zoomathia thesaurus should be linked to TAXREF). On the construction of a SKOS thesaurus of expert zoological knowledge, I co-supervised the work of Irene Pajon, post-doctoral researcher in Classics. She developed the THEZOO thesaurus, in SKOS [80]. It integrates various kinds of knowledge, ranging from zoonyms, to archeotaxa, anthroponyms, toponyms, concepts of anatomy, ethology, physiology, etc., with the aim of enabling the analysis of texts according different perspectives. The development process of THEZOO combined (1) the manual annotation of books VIII-XI of Pliny the Elder's Natural History, chosen as a reference dataset to elicit the concepts to be integrated in the thesaurus, and (2) the definition in natural language of the elicited concepts in the thesaurus and their hierarchical organization. The choice of a convenient organization is quite difficult given the heterogeneity and interrelations of the elicited concepts. The thesaurus is multilingual, combining Latin, Ancient Greek, and three modern languages: French, Spanish and English. The thesaurus is built by using the OpenTheso editor. The annotation was performed in MS Word documents, intended to be transformed into XML and ultimately RDF, by using XSL transformations.

On the construction of a SKOS representation of the TAXREF reference, it has been used as a real-world use case to experiment and evaluate the approach and tool developed by Franck Michel in his PhD thesis [75], which I co-supervised, on the transformation of non relational data into RDF: we transformed the JSON export of TAXREF in SKOS. This is further described in Section 4.1 of Chapter 4. Upstream, we conducted an ontology engineering work with the researcher from the MNHN to discuss the status of each type of object in the TAXREF database and we built a thesaurus model in SKOS-XL which extends SKOS to support the description of lexical entries [20]. In our model, the reference scientific name of a taxon and its synonyms are not literals but URIs (values of properties `skx:prefLabel` and `skx:altLabel` respectively); the label literal values themselves are attached to them as values of property `skx:literalForm`. The habitat and biogeographical status are attached to taxa, while the authorities, taxonomical rank, and vernacular names are attached to labels. We defined a specific TAXREF vocabulary (SKOS concepts) for the taxonomical ranks, types of habitat and biogeographical statuses, aligned with LOD vocabularies.

Not surprisingly, given the complexity of the zoological vocabulary, the first approach experimented to extract zoological knowledge from texts, based on lexico-syntactical patterns, poorly performed, except for zoonyms. In the framework of the Master thesis of Konstantina Poulida and Safaa Rziou, which I co-supervised with my colleague Andrea Tettamanzi, we took advantage of the now available THEZOO thesaurus and we reused state-of-the-art NLP methods and supervised learning algorithms and libraries for the categorization of text segments [56]. The developed classifier is a set of binary classifiers deciding for each considered category whether a segment belongs to it or not. Categories can be any concepts of the THEZOO thesaurus and the semantics of the subsumption relations among concepts are taken into account in our classifier. To overcome the lack of available terminological resources for Ancient languages and take advantage of the amount of terminological resources developed in the community for modern languages, we considered modern translations of ancient

texts. To compensate the possible lost of precision in processing a translation rather than the original text, we considered several modern translations for each ancient text and we combined the results of their processing. Finally, the identified categories are used to annotate the original ancient text. The results of this approach were encouraging but still preliminary.

1.3.2 Ontology-oriented Art Heritage

In the framework of a collaborative project with the FBK ICT center in Italy, related to the VERBO-VISUAL-VIRTUAL project[10] aiming at developing a unified virtual access to the Archivio di Nuova Scrittura (ANS) collection, I co-supervised with my colleagues Elena Cabrio and Serena Villata the Master thesis of Ahmed Missaoui on the construction of an RDF dataset describing the artworks and artists in the ANS collection [19]. For the data model, we reused and extended the EDM[11] ontology. Starting from the metadata available in the database of the MART museum, we converted it in our RDF model. Then we enriched it by linking all the artists and places in the dataset to DBpedia. Finally, since more than half of the artists in the ANS collection are not part of mainstream movements and therefore do not have a DBpedia entry, we further enriched the dataset by applying state-of-the art NLP techniques (syntactic patterns) to extract knowledge from manually selected texts available on the Web.

In the framework of the AZKAR project[12], that focuses on the remote control of a mobile robot using the emerging Web technologies WebRTC for real time communication, I co-supervised with my colleague Michel Buffa the Master thesis of Hatim Aouzal who focused on one of the use cases of the scenarios addressed in this project: the remote visit of the French Museum of the Great War [16]. To support the remote control of a mobile robot in the museum, for designing the visit, selecting locations and orientations in the museum and relevant multimedia resources to propose to the visitors, we developed an ontology to represent museum objects, scenes, points of interest, maps, trails, we built an RDF dataset from the metadata in the Flora relational database of the museum, and we linked it to DBpedia. As a result the remote control of the robot was based on SPARQL queries on this dataset and WebRTC for real time communication.

1.3.3 Ontology-oriented Software Heritage

In the framework of the Desir project[13], I addressed the problem of preserving procedural scientific knowledge in Life Science for data analysis. With my colleague Pascal Neveu, Olivier Corby and Isabelle Mirbel, we initiated a knowledge management action aiming to manage, share, reuse and promote R functions written by researchers and technicians in a multidisciplinary research team in life science at the LEPSE laboratory, specialized on the analysis and modelling

[10] `http://dh.fbk.eu/projects/vvv-verbo-visuale-virtuale-la-piattaforma-di-ricerca-interattiva-dellarte-verbo-visuale`

[11] Europeana Data Model: `http://pro.europeana.eu/edm-documentation`

[12] `http://www.azkar.fr/`

[13] `http://www-sop.inria.fr/edelweiss/projects/desir/wakka.php?wiki=ColorDesirHomePage`

of plant responses and adaptation to variable environmental stresses [26][79]. We designed an OWL ontology to annotate R functions in RDF, so that in the so-built semantic repository, R functions can be retrieved with SPARQL queries. This was the same approach as the one I adopted to capitalize SPARQL queries in the Building Industry, this time applied to R functions management, and for the R community, this was a new contribution. As a result, a new kind of semantic software repository has been developed, based upon the Corese semantic factory and accessible through a Web Service. It provides an environment for the team members to upload and describe their R functions and to retrieve and download the shared R functions. This application was developed in 2010 and is still maintained and used at LEPSE.

In the framework of Oumy Seye's PhD thesis [83], I addressed the problem of preserving inference rules and business rules on the Web again as a classical problem of knowledge management, as soon as rules are considered as resources of epistemic communities. We proposed an approach to publish, share and reuse rules on the Web based on the representation in RDF of both rule annotations and rule contents themselves, while preserving the interoperability of this representation with the W3C recommandation RIF [84]. This allows the publication of this kind of procedural knowledge on the Web of data. Then we developed a library of SPARQL queries to support the management of rule bases throughout their lifecycle: (1) the construction of rule bases for a given context or application, based on the selection of linked rules available on the LOD, (2) their validation with respect to the RDF datasets on which rules are intended to be applied, (3) their update, (4) their exploitation within inference engines which can be optimized based on rule selection with respect to the targeted RDF data sources [85]. We used the Corese semantic factory to implement and evaluate our proposals.

Conclusion

Throughout the overall work presented in this chapter, I implicitly developed methodologies to build ontologies but never made them explicit as scientific contributions in the domain of ontology engineering. This should be considered as future work.

Regarding my early work in the field of e-Education, the main motivation of ontology-oriented modelling in the QBLS and OrPAF systems was to provide learners with a conceptual navigation among digital learning resources or concepts. Such a representation can also serve as a basis to adapt the system to the user context or profile and to help manage the knowledge base, facilitate the discussion among teachers, or learners. This is further discussed in the next chapter. Since 2015, I started again to work in the field of e-Education, this time in an industrial context. As ontology-oriented modelling is now a widespread approach in e-Education, one of the challenges in the two projects I am leading is to develop *reference* ontologies, aligned with the Linked Data, enabling to annotate and integrate heterogeneous learning objects from various sources. This is discussed in the concluding Chapter 4.4.2.

The same observation holds for my projects on Cultural Heritage, targeting the production of reference ontologies, requiring an effective collaborative work with domain experts. During the last decade, the availability of relevant data

sources or vocabularies on the Linked Data has become a reality and one challenge is its extensive and innovative exploitation to provide added-value services tailored to the needs expressed. This is further discussed in Chapter 4.

Regarding my early work on conformance checking in the building industry, it can be viewed as a first step which led me to address the general question of representing constraints and validating RDF descriptions against constraints. In that sense, the work presented in Section 4.3 of Chapter 4 on the validation of RDF datasets against constraints is a continuation of what I presented in Section 1.2. It also answers a key limitation of our early proposal in producing adapted conformance checking reports.

Chapter 2

Modelling Community Members and Social Structures

Introduction

Chapter 1 dealt with capturing the social semantics of epistemic communities into domain ontologies and ontology-based representations of the community knowledge. This chapter summarizes my contributions going a step further in "bridging the gap between social semantics and formal semantics in epistemic communities" [63], addressing the general research question of *How should we model community members, social structures and social interactions in order to improve the management of the digital resources they share?* In some cases, communities are not explicit and the question which arises upstream is *How should we detect communities from social networks and the interests which bind their members?*

The thesis defended here, and more generally the rationale behind the whole research work of the WIMMICS team and the FORUM theme in SPARKS, is that the user plays a key role on Web applications and therefore must be given the best attention in the modelization task: in any Web application, digital resources are *socially* gathered and have an implicit social status that must be taken into account and explicited in their annotation; the user itself must be modelled, her knowledge, her profile, her context, her activity, as non digital but identifyable resources on the Web. Considering the general aim of supporting epistemic communities, not only the digital information resources and knowledge of the community should be modelled, but also the community members themselves and the social structures holding within the community, to integrate and combine these different kinds of knowledge in the reasoning.

Nowadays, taking into account the user profile in knowledge management systems is a well-known key challenge to enhance the user experience. Since 2008, one of the main topics of the International Conference on Knowledge Engineering and Knowledge Management (EKAW) is "Social and Cognitive Aspects of Knowledge Representation". But before 2006, this was not a main

concern in the KRR community. It appears in connection with the social Web and the semantic Web concerns, as it can be shown in EKAW 2014 proceedings [89][64]. In the semantic Web community, modelling users also arises with modelling social structures and social interactions, the topic is present from the first ISWC conference in 2002 — "Socio-cultural and collaborative aspects". It becomes a major concern from 2005, with the advent of the social Web, as the topics "User-centred applications of the semantic Web" and "Social software" show it. The keynote of Tom Gruber at ISWC 2006 [65] is a milestone in the coming together of the social Web and the semantic Web, and from 2007 the so-called social semantic Web is a main topic of ISWC. The birth of the WIMMICS team and its research program centred on modelling users and epistemic communities comes in direct line with the emergence of these challenges in the Knowledge Engineering and semantic Web communities. As soon as 2006, Fabien Gandon discussed the social dimension of the semantic Web at the French national conference on Knowledge Engineering, IC 2006. His habilitation thesis defended in 2008 further discusses it [62]. In 2009, he published at ISWC with his PhD student Guillaume Ereteo the results of their work on the analysis of online social networks using semantic Web frameworks [55].

I first tackled the challenge of modelling users and later on social relations in learning communities to personalize the access to pedagogical resources, the recommendation of pedagogical resources, and social interactions in learning environments. In the early 2000s, the prime importance of taking into account user profiles and user communities in designing a Web system, to enhance the user experience already was a self-evident truth in the domain of e-Education. For instance, this can be observed in the proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA) of these years, where many papers deal with "Learner Centred systems", "Adaptive Learning Environment", "Computer-Supported Cooperative Learning", "Adaptive Educational Hypermedia", "Collaborative Learning", "Teaching Webs", "Learning Communities".[1] This early concern in e-Education originates from Intelligent Tutoring Systems (ITS) in Artificial Intelligence in the 1970s. [86]

Relatedly, I addressed the question of access rights management in community websites (wikis), by modelling the social structure in the community. More recently, I dealt with community detection and community analysis in question and answer websites, for community analysis, expert users detection and question routing, again by modelling the social structure. The ever rising emphasis on these research questions goes along with the present exponential growth of user generated content on the Web.

As an invariant, my approach consists in somehow reifying both a community as a whole and its members, and considering yet another kind of resources: the reflexive knowledge of the community about itself, which should be made explicit and modelled to improve the community management. It is represented with the same graph-based and semantic Web KR formalisms used to represent the rest of the community resources, thus enabling further reasoning capabilities.

This chapter is organized as follows: Section 2.1 presents my work on ontology-based modelling and management of the individual dimension of users. Section 2.2 presents my work on ontology-based modelling and management of

[1] http://www.editlib.org/j/EDMEDIA

communities and community members.

The works synthesized in this chapter have been published in the proceedings of several international conferences and workshops: *World Conf. on Educational Multimedia, Hypermedia & Telecommunications* (ED-MEDIA 2006) [43], *Int. Conf. on Human Centred Software Engineering* (HCSE 2010) [14], *Int. Conf. on Web Information Systems and Technologies* (WEBIST 2011) [13], *ISWC Workshop on Semantic Personalized Information Management (SPIM 2011)* [4], *Int. Conf. on Advances in Social Networks Analysis and Mining* (ASONAM 2014) [73], *Int. Conf. on Web Intelligence* (WI 2015) [18][71], in several international journals: *Journal of Web Semantics* [17], *Int. Journal of Web-Based Learning and Teaching Technologies* [96], *Int. Journal of Web-Based Learning and Teaching Technologies* [97], *Social Network Analysis Mining* [74], and in two book chapters: *Advances in Knowledge Discovery and Management* [15] and *Security and Privacy Preserving in Social Networks* [94].

2.1 Modelling the Individual Dimension of Users

Nowadays, taking into account the user profile, context, preferences, needs in knowledge management systems is a well-known key challenge to enhance the user experience. User-centred software engineering is a widely spread research topic in computer science and among the main research topics of the semantic Web community stand *personalized access to semantic Web data and applications* and *user interfaces and interaction with semantics and data on the Web.* I first tackled the question of *How should we model the individual dimension of users?* in learning environments to personalize the access to pedagogical resources, to adapt the resource recommendation to each individual profile. Later on, I addressed the same question for for a system supporting application composition, where the composition is driven by user preferences in terms of user interface, and for a question answering system where the individual dimension of the user stands in his/her questions.

2.1.1 Modelling Learners within Adaptive Learning Environments

In the domain of e-Education, I addressed the research question of *modelling the user based on ontologies and reason on this model* as a special kind of knowledge which can be combined with domain knowledge or pedagogical knowledge to improve the management of knowledge-based e-learning systems.

Towards Adaptive Learning in QBLS

In the framework of Sylvain Dehors's PhD thesis [42] already discussed in Chapter 1, we started exploring the use of a basic user model to improve the learning experience within an e-learning system. We first defined a simple RDFS model for expressing a log entry: each log entry is associated with a user, a time stamp, a domain concept and a visited pedagogical resource relative to this domain concept. Then we combined the conceptual graph representing a course and the log entry model to define a graph navigation model where each step of the navigation, i.e. each jump from one resource to another, is decomposed into

a link from one resource to a concept and a link from this concept to a resource relative to it, and reified and linked to the user and the time stamp. At that time, the originality of this proposal was to combine pedagogical resources and concepts on the same graph, enabling to access the *conceptual* level of the user navigation history.

This graph navigation model was viewed as a basic student model which first enabled some adaptations of the system to the user. In the QBLS interface, different colors were associated to the resource tabs, depending on whether a resource had already been visited, is currently visited or had never been visited. Also, we proposed a conceptual Back button, enabling to directly jump to previously visited concepts, avoiding to navigate to all the visited resources related to a same concept. These interface adaptations were the results of dynamically querying the student model represented in RDF with SPARQL queries.

The learning itself was not adaptive, i.e. the proposed conceptual graph was the same for every student. However we defined a simple user profile model capturing pedagogical preferences, e.g. `Fundamental` resources having priority upon `Auxiliary` ones and a set of rules exploited the pedagogical ontology to complete this knowledge: for example `Definition`, a subtype of `Fundamental`, is inferred to be prior upon `Illustration` which is a subtype of `Auxilliary`. We used this user model to rank the learning resources proposed to the user at each navigation step, depending on their pedagogical role [43]. But the same pedagogical preferences were used to set the QBLS interface for every student. In that sense it was not really a user profile model but still a first step towards it.

Relatedly, we experimented querying the RDF base of graph navigation models with SPARQL queries for the teacher to retrieve and visualize the model of a chosen student, or an aggregated view of all the student models (for instance to detect regular characteristics of the course, e.g. most visited concepts and resources, resources never accessed, etc.), or to compare student models, e.g. the navigational models of the students who never access example resources related to the concepts they explore, or the student who tend to visit answers to queries before accessing the course, those who only use answers afterwards, students who perform a "clean" navigation, students who need many iterations on the same material, etc. Figure2.1 shows the navigation path of a student within the conceptual graph. The path used by the student is highlighted in red; the width of the edges is proportional to the number of times the learner has performed this step. Such semantic Web based visualization of the semantic course structure and student models was aimed at supporting their behavioural and cognitive interpretation by the teacher, to help him/her adapt his/her teaching strategy.

Adaptive Learning in OrPAF

In the framework of Amel Yessad PhD thesis [95], already discussed in Chapter 1, I went a step further in taking into account a user model to develop an adaptive learning environment. In OrPAF, we focused on modeling the user of a learning organizer to provide him with adaptive learning paths according to his model [96]. The learner model captures the learning goal, the level of knowledge, which depends on the results of the tests passed by the learner during the learning process, and which evolves over time, and preferences (preferred resource

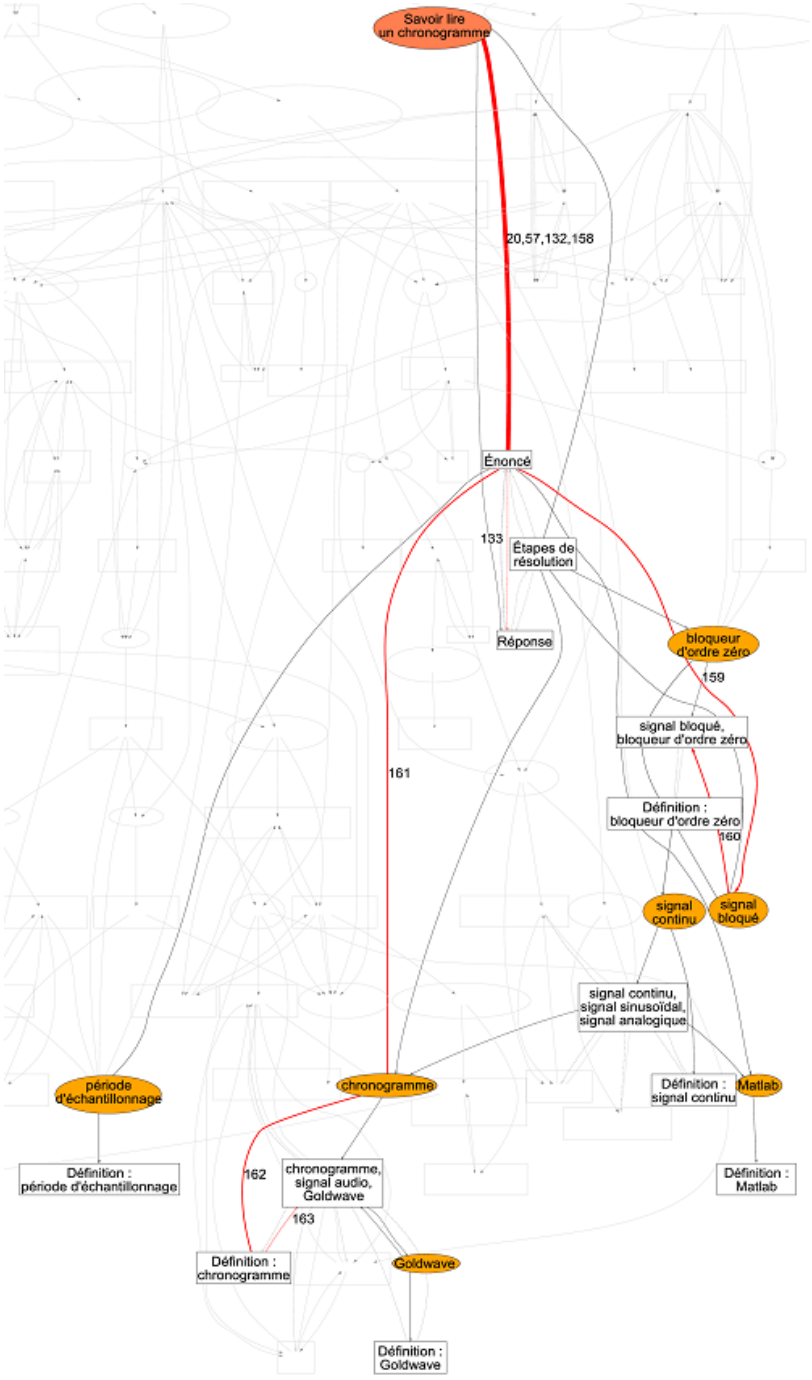

Figure 2.1: Navigation path of a student from the starting question of how to read a digital timing diagram

language, preferred author, preferred format, cognitive type) of the learner. As a result the learning organizer provides a mechanism for self regulated learning.

In our approach, the structure of the course presented to the learner is an adaptive conceptual map, a sub-graph of the graph of concepts building up the domain model. We use the graph structure of the domain model as a roadmap to generate learning paths. Given a certain goal concept that the learner wants to acquire (e.g. `Statement`) and given his learner model, the learning organizer filters the domain model and generates a map of learning concepts that the learner must learn to achieve his/her goal. Figure 2.2 presents the cognitive map of a learner in the domain of algorithm and programming, with `Statement` as goal concept.

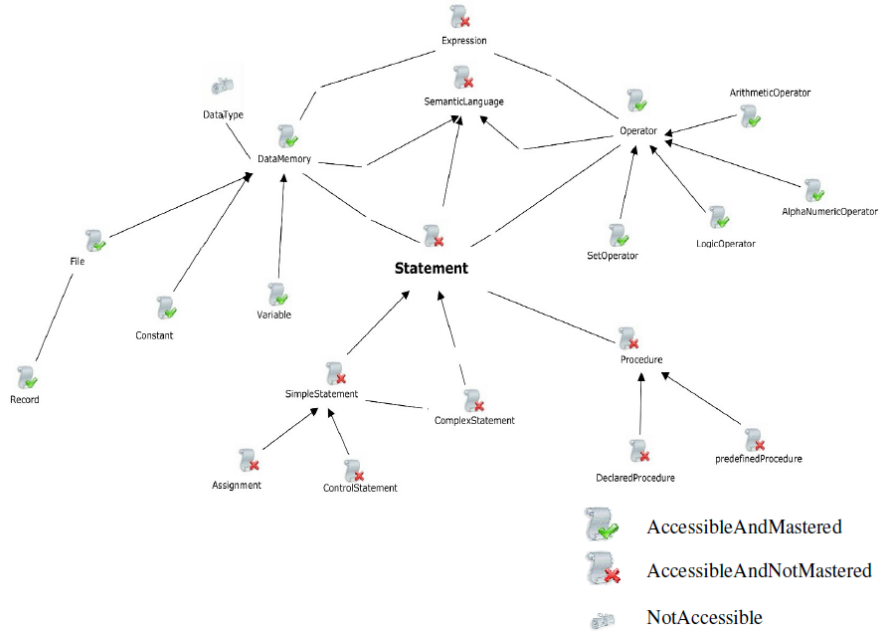

Figure 2.2: The cognitive map of a learner in the domain of algorithm and programming, with `Statement` as goal concept

Depending on the tests already passed by the learner, each concept on the map is indicated as being mastered (e.g. `Operator` or not; for non-mastered concepts, they are indicated as being accessible (`Statement`) or not (`DataType`), depending on their relations with the other concepts of the map: to be accessible, all the prerequisites of a concept must be mastered by the learner. The system can generate three types of cognitive maps depending on the time constraints of the learner: a simple map comprising the domain concepts related to the goal concept by transitive closure of the prerequisite relationship, a hierarchical map extending the simple map with the specializations and generalizations of the goal concept, a relational map extending the simple map with all the concepts related to the goal concept by a path of relationships of length lower than a given threshold.

Not only the structure but also the content of the course is adapted to the learner: the learning resources proposed to the learner when clicking on an acces-

sible concept of his/her map are selected from the local repository depending on their relevance to his preferences and his learning context, i.e. his/her learning goal and the current state of his/her conceptual map [97]. We defined a measure for calculating the semantic relevance of a learning resource for a specific learning context, based on relative weights assigned to the domain concepts occurring in the annotation of the learning resource. The relative weight of a domain concept depends on the length of the path to reach it from the current concept of the learner in his/her conceptual map and the type of relationships in the path: the weight of a specialization is greater than the weight of a prerequisite which is greater than the weight of an aggregation. The semantic relevance of a learning resource is computed as the sum of the relative weights of the concepts occurring in its annotation and accessible in the map of the learner, weighted by the sum of the relative weights of the concepts occurring in its annotation but not accessible in the map of the learner. The resources which semantic relevance is above a chosen threshold are then further filtered by conducting a similar process with the concepts of the pedagogical model occurring in their annotations: the resources having a pedagogical type (e.g. exercise) similar to the current resource of the learner, i.e. close in the pedagogical domain, will have the highest pedagogical relevance. In a final step, the filter considers the preferences of the learner (e.g. resource language).

2.1.2 Modelling User Interface Preferences to Compose Applications

I participated to the launching stage of the PhD thesis of Christian Brel [12], directed by Michel Riveill, which addressed the research question of *How can we compose an application centred on user preferences.* To answer this question, we proposed an approach of application composition driven by user interface (UI) composition, assuming that user interfaces are a kind of user preferences [14][13]. Our proposal relied on an ontology to represent an application from a multi-faceted perspective integrating tasks, functionalities and user interfaces, and to connect the three. The ontology comprised classes and properties to model an abstraction of usual layouts used in graphics libraries of programming languages, UI components, tasks that can be performed by users and links between UI components, functionalities and tasks. This enabled us to define a semi-automatic UI composition process to assist the developer in composing an application according to user (interface) preferences. This process relies on the ontologies developed, ontology-based inference rules enabling to deduce relative layout of UI components from any layout description, and constraints to preserve the consistency of the user interface being composed.

2.1.3 Modelling User Needs in Question Answering Systems

During the last decade, the popularity of e-commerce has tremendously grown, leading to the emergence of a set of dedicated Business-To-Client (B2C) services and applications, among which Question Answering systems. At the same time, user needs are getting more and more complex and specific, especially when it comes to commercial products, with questions related to their technical characteristics. One of the challenges this raises is *How can a system understand and*

interpret complex Natural Language (NL) questions in a specific (commercial) context? Modelling NL queries in a Question Answer system can be viewed as a special case of user modelling, where the user is represented by his queries.

To answer this research question, in the framework of a collaborative project of which I am the scientific leader, with the SynchroNext company, we proposed an ontology-based approach to understand and interpret complex NL questions in a specific domain, and answer it by querying RDF datasets. The final aim was the conception of a chatbot specialized in a given e-commerce domain. We experimented in the domain of the mobile phone industry: we modelled a dedicated RDFS ontology and we built the QALM RDF dataset by extracting raw data from eBay and BestBuy commercial websites and transforming it into RDF [66]. The question interpretation relies on a state-of-the-art approach to identify the key elements in the question — namely the type of the resource looked for (the Expected Answer Type (EAT)), the Named Entities (NE), and the properties linking named entities between them or to literal values —, and build a graph representation of it — by connecting the identified triples. The identification of property values and of properties themselves connecting entities relies on the use of domain-dependent regular expressions (regex). The originality of the proposed approach lies on the fact that these regex are automatically learned from a subset of our dataset with a genetic programming algorithm [18].

The focus of the collaborative project with the SynchroNext company slipped from e-commerce to the insurance field, and we now are working on the automatic categorization of NL questions, instead of their interpretation and formalization: the analysis of database of messages provided by the company showed that the client messages to their insurance company are much longer than in e-commerce sites and the questions hardly are explicit. Here again we rely on the exploitation of terminological resources to build a vector representation of the questions to be processed with state-of-the-art machine learning algorithms. I co-supervise the PhD thesis of Raphaël Gazzotti on this subject, with my colleagues Fabien Gandon and Elena Cabrio.

2.2 Modelling the Social Dimension of Community Members

In community websites, the social dimension of the user model becomes a key feature to improve the management of the shared resources. I addressed the question of *How should we model the social dimension of users?* in three different domains: in a collaborative website to manage access rights to the community resources; in a learning environment to enhance resource recommendation based on peer experience; and in a question answering site to support the community management throughout its life-cycle.

2.2.1 Ontology-based Access Rights Management in Collaborative Websites

Access control represents a major challenge in content management systems and is central to collaborative Web sites where collaborative editing of documents and sharing raises the question of the definition of access rights. With my

colleague Michel Buffa, we proposed an ontology-based approach to manage access rights in collaborative websites [15][94]. The ontology models agents, roles, actions and access types; it is used to annotate the resources of the website (community documents and community members) and to declaratively represent a control strategy as a set of inference rules, specifying the rights granted to agents of a given resource, or describing general access laws, e.g. stating that a member of a group inherits the roles assigned to her group, or that the creator of a resource is an agent of this resource. The access management for the annotated resources is then based on reasoning on the RDF dataset of semantic annotations, with the ontology-based inference rules (expressed as SPARQL queries of the CONSTRUCT form) and querying it with SPARQL queries (of the SELECT form) to retrieve knowledge about the authorized access to a specific user on a given resource. We applied our approach for access right management in the semantic wiki SweetWiki [17].

2.2.2 Folksonomy-based Resource Recommendation

Social tagging and the resulting folksonomies enable to collaboratively explicit and share knowledge about the resources of a community. This can be used to organize and access resources based on the tags annotating them and capturing their social semantics. I started addressing the question of social recommendation of resources with my colleague Hassina Seridi, in the framework of Samia Beldjoudi's PhD thesis [3]. More precisely, we addressed the research question of *How can we develop a folksonomy-based recommender while overcoming the well-known limitations of folksonomy-based access to resources, due to the lack of semantics of tags?*

To answer this question, we proposed an approach based on the exploitation of association rules extracted from the social relations in a folksonomy, in order to recommend to the user not only resources annotated with the same tags than the ones he/she uses but also other resources annotated by other users close in the social network, with other tags suggested by association rules [4].We argue that the automatic sharing of resources strengthens social links among actors and we exploit this idea to reduce tag ambiguity in the recommendation process by increasing the weights associated to resources according to social similarities.

More precisely, we represent each user in a folksonomy by a transaction ID and the tags he uses by the set of items which are in this transaction; then we use the state-of-the-art *A priori* algorithm to extract association rules between sets of tags. For each extracted association rule which applies to the current user, i.e. whose antecedent is a set of tags used by the current user, the resources tagged with the tags found in the consequent of the rule are candidate to be recommended by the system. The effectiveness of the recommendation and the ranking of the recommended resources depend on the similarities between the current user and the other users who use the tags occurring in the consequent of the rule. The similarity between two users is measured by the cosine similarity between the vectors of tags they use.

We applied our approach to support the social recommendation of resources on diabetes within a community of medical interns, to help them acquire the best practices to patient diseases diagnosis and treatments. The system developed can be viewed as a kind of adaptive informal learning environment, where the social profile of learners is exploited to adapt the recommendation of pedagogical

resources.

2.2.3 Community Detection in Question Answer Sites

In the framework of Zide Meng's PhD thesis [70], which I co-supervised with my colleague Fabien Gandon, we considered again the exploitation of social tagging, here with the aim of detecting topics from tags and, based on them, detecting user communities.

In many social networks, people interact based on their interests. Community detection algorithms are then useful to reveal the sub-structures of a network and in particular interest groups. Identifying these communities of users and the interests that bind them can help us assist their life-cycle. However certain kinds of online communities such as community question answering (CQA) sites have no explicit social network structure and many traditional community detection techniques do not apply directly. Therefore we addressed the research questions of *How can we detect communities of interests in CQA sites and how can we identify the common topics that bind them?* Figure 2.3 presents an overview of the work conducted within this thesis.

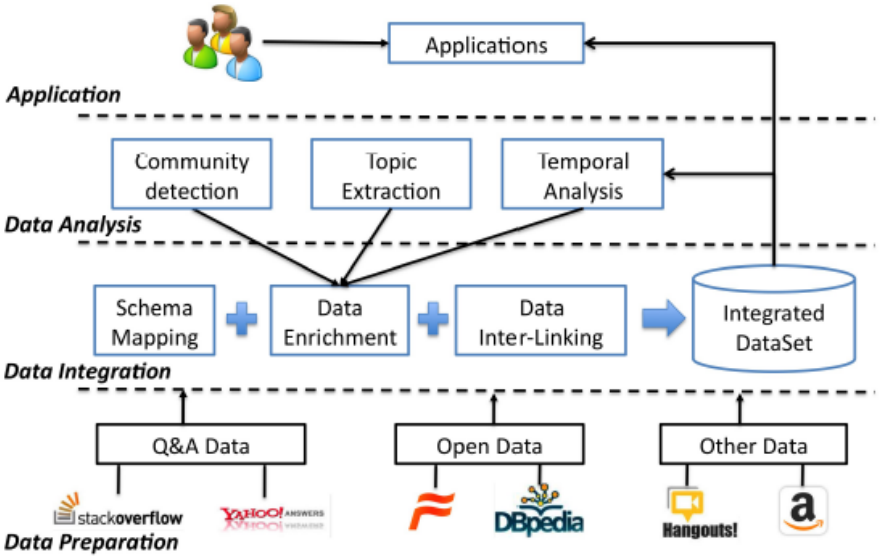

Figure 2.3: Overview of the proposed framework to analyse QA site content and communities

We proposed an approach for extracting topics from Q&A's tags to detect communities of interest and we applied it on a dataset extracted from the popular CQA site StackOverflow [73][71][74]. Broadly speaking, our approach consists in first detecting topics based on question tags; these topics are then viewed as communities of interests and users are assigned to them depending on the tags of the questions they answer; by construction, these communities of interests are overlapping when users are interested by several topics.

As a preliminary step, we developed an RDFS vocabulary reusing and extending SIOC and FOAF in order to formalize in the semantic Web standards

within a single unified model both the knowledge extracted from the raw user-generated content of the social media platform (questions, answers, users, votes, comments), and the latent knowledge constructed by mining and reasoning on this explicit knowledge (topics, interests, expertises, activities, trends). Then we converted a native StackOverFlow dataset into this model and aligned the tags with DBpedia [72].

Based on this representation model, we first experimented the state-of-the-art Latent Dirichlet Allocation (LDA) model to answer the above described problem of community detection. The LDA model is commonly used in NLP to explain observations of words in documents by *latent*, i.e., unobserved, topics, and then assign these topics to the documents. We used it to assign topics to users based on the tags attached to questions in CQA sites, by considering users and tags just like documents and words: tags attached to questions are acquired by users when answering the questions. Community detection is here considered as a clustering problem where users with similar topics of interest are grouped into the same cluster forming a community of interest.

We showed the limitation of this approach, in particular in terms of complexity, and we proposed a much simpler and more efficient method, based on the observation of our dataset, which empirically confirmed the natural intuition that high frequency tags are more generic and low frequency tags are more specific, that most of the low frequency tags are related to a more generic tag, and that the first tag of the tag list associated to a question normally is much more generic than the others and indicates the domain of the question. Based on this observation, our approach consists in building a set of tag trees according to the position of tags in the tag lists associated to the questions, i.e., one tree for each of the tags encountered at least once at the first position within a tag list, these tags being the roots of the trees. Figure 2.4 shows the tag tree for tag *html*.

We then construct an affinity matrix of the root tags where the similarity of two tags depends on their numbers of occurrence and co-occurrence and we perform a spectral clustering on this matrix to group these root tags, each group forming what we call a topic. Finally we combine the trees whose roots belong to the same topic. As a result we get a forest of trees, each tree representing a topic, and for each topic tree we compute the probability for a tag to belong to the topic, thus producing the topic-tag distribution. The user-topic distribution is then easily computed by representing each user by the list of tags associated to the questions he/she answered and by computing the list of probabilities that he/she is interested in each topic as the sum of the probabilities his/her tags belong to the topic. The resulting user-topic distribution, as depicted in 2.5, enables to easily identify the users' communities of interests: a user having a high probability for a topic should be a member of the community represented by this topic.

The observation of our dataset showed that one third of the questions only have one or two tags and that, in this case, the tags are less popular and the main domain is implicit. Therefore, to enable the application of the above described method to all the questions, we added a preliminary step to enrich a question with a first tag when needed. The approach is as follows: A first-tag distribution is computed associating each tag to a list of candidate first-tags with estimated probabilities. Then a first-tag is chosen to enrich each list of tags associated to a question as follows: given a tag list, its top 5 candidate first-tags

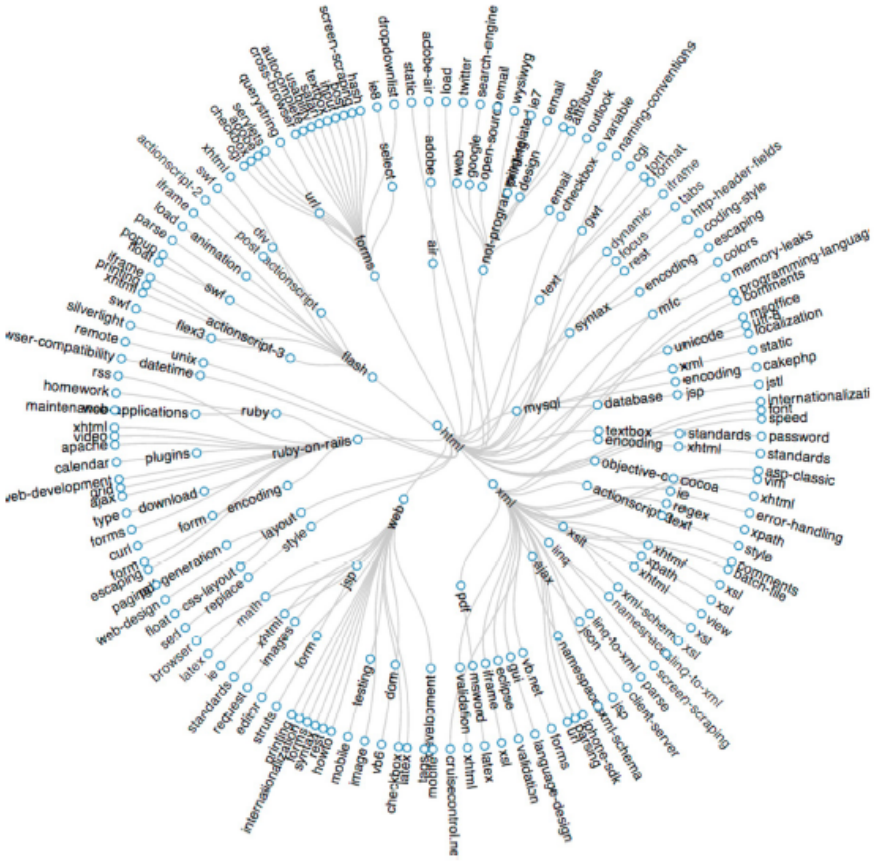

Figure 2.4: Html tag tree

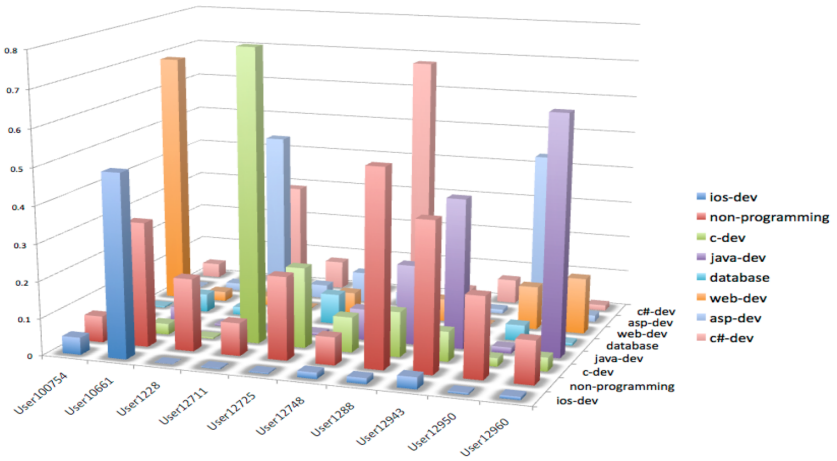

Figure 2.5: User-topic distribution

(with the highest probabilities) are fetched, their corresponding probabilities are cumulated with a discount function depending on the position of the associated tag in the tag list, and the candidate first-tag with the highest probability is inserted at the first position of the tag list (unless it already belongs to the list). This enrichment of the tag lists significantly reduced the number of tag trees to be generated, enabling our method to scale.

To support user interactions, we considered automatically generating labels to turn bags of words (tags) into meaningful, i.e. the communities of interests. Our approach relies on DBpedia as external knowledge to help with choosing labels. Tags are linked to DBpedia resources, and the graph of related resources in DBpedia with relation paths of length smaller or equal to a chosen threshold is extracted to select topic labels among the resource labels. A user survey showed that users can reach a good agreement on composite labels, the best agreement score was achieved with a combination of the top 3 voted labels. To automatically generate these composite labels, we proposed a hybrid solution combining the results of different graph algorithms/metrics, ranking them, and selecting the top 3 ones.

Finally, we generalized the LDA model initially tested to detect topics from question tags, and we proposed a joint probabilistic graphical model to extract topic-based expertise and temporal indications from Q&A sites, with the ultimate goal of supporting question routing, expert recommending and community life-cycle management. We not only consider question tags, but also answer posts inheriting question tags, words in answer posts, votes on answers, and timestamps.

Conclusion

In the coming years, among the research focuses tackled in this chapter, both the modelling of community members and social structures and the modelling of user needs expressed in natural language should take a growing part in my work. The collaborative project EduMICS starting this year with the Educlever company specialized in the design of learning solutions for primary and secondary schools opens up the opportunity to analyse, model and reason on a large base of real world user profiles, traces, interactions. This should ensure the continuation of my work in modelling the individual and social dimension of learners, but also the exploration of the possibilities to adapt my work on community detection for Q&A sites. The collaborative project starting this year with the SILEX company specialized in B2B solutions for linking service providers and clients may also be an opportunity to adapt my work on Q&A sites, by considering service offers as answers to service requests.

As for the modelling of user needs expressed in natural language, in addition to the ongoing collaboration with the Synchronext company on the analysis and categorisation of client questions in natural language in the insurance domain, the collaborative project with SILEX will also provide real-world use cases: the linking of service providers and clients will require analysing and modelling the service offers and requests expressed in various natural languages. Relatedly, the collaborative project starting this year with the Gayatech company aims at generating educational quizzes — questions and answers — in natural language from the Linked Data.

Chapter 3

Graph-based Knowledge Representation and Reasoning on the Semantic Web

Introduction

This section synthesises my scientific contributions related to formal semantics on the Web, addressing the general research question of *How to represent knowledge and perform reasoning on the semantic Web?*, with the strategic bias of a *graph-based* approach to KRR, justified by the graph nature of the Web.

The semantic Web aims at providing models and techniques to represent knowledge on the Web and reason on it. As a result, the Web has developed a new facet, a giant knowledge base exploited both by human and software agents. To realize this semantic Web, the challenge during the last 16 years was to provide KRR models compatible with the Web architecture and specificities (e.g. open world assumption). As I started working in this domain in 2000, the RDF model had just been recommended by the W3C a year before. The adoption of this standard and its implication for the Web were still to come. During my PhD in the 90s the Web was exploding as a network of interlinked *information* pieces readable by human beings. In the meantime, there were a wide range of works on hypertexts and hypermedias, some of them in Artificial Intelligence, dealing with bringing semantics to them. My PhD thesis dealt with the construction of hypermedias based on the indexation of the elements of these systems by formal *knowledge* pieces, *semantic annotations*, which it was possible to reason on. These were the very same principles underlying the upcoming semantic Web.

At that time, there were two main trends, two main research communities and therefore two main competing models for declarative knowledge representation in Artificial Intelligence: Description Logics and Conceptual Graphs, both descending from Semantic Networks. Description Logics emphasized logical reasoning while Conceptual Graphs defended graph-based reasoning. Description

Logics early imposed themselves for KRR on the semantic Web: OWL, the language for formalizing Web vocabularies recommended in 2004 belongs to this family of languages. This can be explained by the fact that the Description Logics community immediately adopted the vision of a Web-oriented KRR and invested time and people in W3C activities. However, a minority trend early defended a graph-based formalization of the semantic Web and RDF is indeed a graph model. This could be seen in the Web Design Issues[1] behind the W3C recommendations written by Tim Berners Lee in 1998. This is nowadays an evidence, with the advent of the notions of *Giant Global Graph* and *Web of Data* both introduced by Tim Berners Lee in 2006. The *Web of Data* can be defined as the first successful deployment step of the semantic Web, limited to RDF and RDFS for the formalization of data and vocabularies, and SPARQL for querying (and excluding OWL). The *Knowledge Graph* introduced by Google in 2012 is based on the same principles of the Web of Data with proprietary access and model.

The ACACIA team early adopted this graph-based view of the semantic Web. As I joined the team in 2000, Olivier Corby and Rose Dieng Kuntz had just published a paper on the correspondance between the Conceptual Graph model and RDF model [23]. I had myself acquired a good knowledge and practice of the Conceptual Graph model during my PhD thesis and I adhered to this view straight away. From that time, I addressed the research question of graph-based and semantic Web KRR. Not surprisingly, KRR appeared among the topics of the first International Semantic Web Conference (ISWC) in 2002, and conversely, the same year, the semantic Web was one of the track of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW).

This chapter is organized as follows: Section 3.1 is dedicated to my early works in the use of the Conceptual Graph model for KRR on the semantic Web. Section 3.2 shows the evolution of my perspective to a *generic* graph-based model for the semantic Web. Section 3.3 is dedicated to my works on KRR with graph rules on the semantic Web.

The works synthesized in this chapter have been published in the proceedings of a national French conference: *Langages et modèles à objets* (LMO 2010) [31], in a French journal: *L'Objet* [52], and in the proceedings of several international conferences and workshops: *Int. Conf. on Conceptual Structures* (ICCS 2001, ICCS 2007, ICCS 2008) [51][28][1], *Int. WWW 2002 Workshop on Real World RDF and Semantic Web Applications* [27], *European Conf. on Artificial Intelligence* (ECAI 2002, ECAI 2004) [48][24], *Int. Conf. on Web Intelligence* (WI 2007, WI 2010, WI 2012) [29][30][39], *MICCAI 2012 Workshop on Data- and Compute-Intensive Clinical and Translational Imaging Applications* [60], *ECAI 2012 Workshop Artificial Intelligence meets the Web of Data* (AImWD 2012) [84], and in an international journal: *IEEE Intelligent Systems* [25].

3.1 Conceptual Graphs for the Semantic Web

This Section is dedicated to my early works dealing with the question of *How can the Conceptual Graph model be used for KRR on the semantic Web?*. Starting from the correspondence established in [23] between the Conceptual Graph

[1] `http://www.w3.org/DesignIssues/RDFnot.html`

model and the RDF model, I started to work in 2001 on KRR in the RDF model inspired by the Conceptual Graph model. In [51], [27], and [52], my co-authors and I highlighted the similarities and correspondences between CG and RDF models. With Alexandre Delteil, I addressed the problems of representing contextual knowledge and representing definitional knowledge in the RDF model — this was before the recommendation of the OWL standard in 2004. Alexandre Delteil was a PhD student directed by Rose Dieng Kuntz and this was my first experience in participating to the supervision of a PhD. At the same time, I started to work with my colleague Olivier Corby on querying RDF knowledge — this was long before the recommendation of the SPARQL query language for RDF in 2007. A few years later, in 2007, I worked again with him on representing and reasoning on contextual knowledge.

Section 3.1.1 presents the result of our comparative study of the Conceptual Graph model and the RDF model, and the following sections present our work on KRR in the RDF model based on features taken from the Conceptual Graph model: Section 3.1.2 presents our work on the representation of contextual knowledge; Section 3.1.3 presents our work on the representation of ontological knowledge; Section 3.1.4 presents our work on querying RDF data.

3.1.1 RDF and Conceptual Graphs

This section summerizes the similarities and correspondances between CG and RDF models which we highlighted and detailed in [51], [27], [52].

Resource Description Framework (RDF)

RDF has been created in 1997 as a working draft and was recommended by the W3C in 1999. An RDF description consists in a set of *statements*, each one specifying the value of a property for a resource. A statement is thus a triple `(resource property value)`, where `resource` is a URI identifying a resource, `property` is a URI identifying the type of the property, and `value` is either a URI identifying the property value or a literal. A set of statements can be viewed as a directed labelled graph: a vertex is either a URI or a literal and an arc between two vertices is labelled by a URI. Blank nodes denote unknown or *anonymous* resources, which cannot be identified by a URI. This corresponds to the expression of existential quantification.

RDF Schema (RDFS) is dedicated to the specification of schemas representing the ontological knowledge used in RDF statements. RDFS has been created in 1998 as a working draft and was recommended by the W3C in 2004. A schema consists in a set of class and property declarations; classes are resources declared as instances of the `rdfs:Class` resource, and properties are declared as instances of the `rdf:Property` resource. Instantiation relations between instances and classes are expressed with the property `rdf:type`. The *signature* of a property consists in one or several *domains* and one single *range*: the domains of a property can be used to type the subject of the triples using this property, and its range to type the value of these triples. Properties `rdfs:subClassOf` and `rdfs:subPropertyOf` enable to define class hierarchies and property hierarchies.

Conceptual Graphs (CG)

The Conceptual Graph model has first been defined in 1984 in [87]. A conceptual graph is a bipartite (not necessarily connected) graph composed of *concept nodes* and *relation nodes* describing relations between these concepts. Each concept node c of a graph G is labelled by a couple $< type(c), referent(c) >$, where $referent(c)$ is either the *generic marker* * corresponding to the existential quantification or an *individual marker* corresponding to an identifier; M is the set of all the individual markers. Each relation node r of a graph G is labelled by a *relation type* $type(r)$; each relation type is associated with a *signature* expressing constraints on the types of the concepts that may be linked to its arcs in a graph.

Concept types (respectively relation types of same arity) build up a set T_c (resp. T_r) partially ordered by a generalization/specialization relation $\geq$. (T_c, T_r, M) defines the *support* upon which conceptual graphs are constructed. A support thus represents the ontological knowledge.

The semantics of the Conceptual Graph model relies on the translation of a graph G into a first order logic formula thanks to a ϕ operator such that $\phi(G)$ is the conjunction of unary predicates translating the concept nodes of G and n-ary predicates translating the n-ary relation nodes of G; an existential quantification is introduced for each generic concept.

Conceptual graphs are provided with a *generalization/specialization relation* $\leq$ corresponding to the logical implication: $G_1 \leq G_2$ iff $\phi(G_1) \Rightarrow \phi(G_2)$. The fundamental operation called *projection* enables to determine the generalization relation between two graphs: $G_1 \leq G_2$ iff there exists a projection π from G_2 to G_1. π is a graph morphism such that the label of a node n_1 of G_1 is a specialization of the label of a node n_2 of G_2 with $n_1 = \pi(n_2)$. Reasoning with conceptual graphs is based on the projection, which is sound and complete with respect to logical deduction.

Projection is sound and complete with respect to logical deduction. Finding a projection between two graphs is a NP-complete problem.

Correspondences between the RDF Model and the CG Model

As it appears in their descriptions, CG and RDF models share many common features. Both models distinguish between ontological knowledge and assertional knowledge. In both models, the assertional knowledge is positive, conjunctive and existential and it is represented by directed labelled bipartite graphs (bipartite in the sense that we can reify properties and distinguish between graph nodes representing properties and graph nodes representing the subjects and objects of properties). Regarding the ontological knowledge, the class (resp. property) hierarchy in an RDF Schema corresponds to the concept (resp. relation) type hierarchy in a CG support. RDF properties are declared as first class entities like RDFS classes, in just the same way that relation types are declared independently of concept types in a CG support. This is this common handling of properties that makes relevant the mapping of RDF and CG models. In particular, it can be opposed to object-oriented language, where properties are defined inside classes.

There are some differences between the RDF and CG models in their handling of classes and properties. However they can be quite easily handled

when mapping both models. Mainly, the RDF data model supports multi-instantiation whereas the CG model does not and an RDF property declaration may specify several constraints for the domain (resp. range) whereas in the CG model, a relation type declaration specifies a single constraint for the domain (resp. range). However, the declaration of a resource as an instance of several classes in RDF can be translated in the CG model by generating the concept type corresponding to the most general specialization of the concept types translating these classes. Similarly, the multiple domain (resp. range) constraints of an RDF property can be translated into a single domain (resp. range) constraint of a CG relation type by generating the concept type corresponding to the most general specialization of the concept types constraining the domain (resp. range) of the property.

Extension of RDF based on the CG Model

Based on the similarities between CG and RDF models, Alexandre Delteil, Rose Dieng Kuntz and I proposed an extension of RDF based on features of the CG model. Regarding the similarities of the RDF(S) and CG models, we argued in 2001, the early age of the semantic Web, that it was a real challenge and opportunity for the CG community to contribute to the elaboration of a standard language for knowledge representation, interoperability and reasoning on the semantic Web — just like the community of Description Logics was doing. Such a mobilization did not happen in the CG community, and our work presented in the following did not have any direct continuation. The proposition of a W3C member submission defining the OWL profile that can be implemented with the CG model could still be considered as future work. The Corese engine (see Section 3.1.4) which stands among the first implementations of RDF/S and prototyped the upcoming SPARQL standard, was based on the CG model, using the Notio API for Conceptual Graphs. More generally, the interest of our work lies in a historical perspective: we participated to a collective pioneer movement aiming at answering research questions which were of prime importance at that time — *Which model enables to represent ontologies on the semantic Web?* and *How can we represent and reason on contextual knowledge on the open Web?*, which were a reactualization of former research questions in the KRR communities and still are research topics.

3.1.2 Extensions of RDF to Represent Contextual Knowledge

In the original RDF model (1999) all statements can be viewed as building up a giant single knowledge graph. On the contrary, in the CG model, assertional knowledge is a base of conceptual graphs built upon a support, and each graph can be viewed as a special context. In 2001, we proposed an extension to RDF to express independent pieces of knowledge, quotations, viewpoints, etc., by reifying a context and explicitly representing and describing it in the whole RDF graph [51]. In 2007, we proposed another extension to RDF to represent and reason on relations between contexts [29]. This time, it was the relation between graphs which was reified. Both extensions were based on similarities between the RDF and CG models. They are described in the following two sections. Of course, these works must be considered in their historical context:

since then, the notion of *named graphs* has been introduced in the RDF model and is part of the RDF 1.1 recommendation.

Reification of a Context

In 2001, we envisioned a context as a mean to identify a subgraph in the whole RDF graph, i.e. to identify or cluster RDF statements belonging to it [51]. Basically, to extend RDFS with contexts, we introduced the class `:Context` and the properties `:isContextOf` and `:referent`. A context is a resource of type `:Context`; an anonymous resource is linked by an `:isContextOf` property to the context it belongs to and by a `:referent` property to the identified resource it refers to in this context. The introduction of this property was directly inspired from the *referent()* function in the CG model. As a result, a context is defined from a resource G of type `:Context` as the largest subgraph of the whole RDF graph whose all internal nodes except G are anonymous resources, values of a `:isContextOf` property with G as subject. For instance, Figure 3.1 presents an RDF graph embedding two contexts. A context is thus an abstraction that enables to talk about resources refered to with anonymous resources rather than directly about resources. A resource can be referred to by several distinct anonymous resources in different contexts. Anonymous resources are externally identified by the referent property. The rules for constructing RDF contexts are based on the translation of conceptual graphs into RDF; they are detailed in [51].

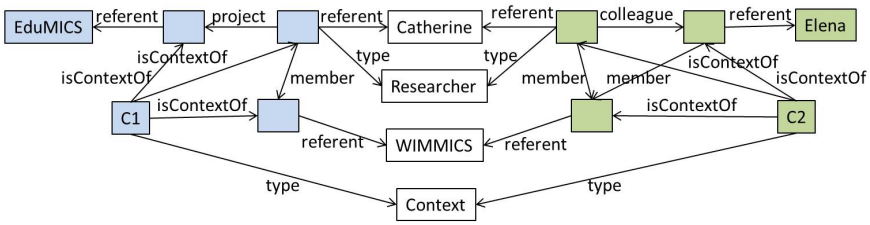

Figure 3.1: Example of an RDF graph embedding two contexts coloured in blue and green; the shared nodes are colourless

The introduction of the `:referent` property also provided the RDF model with a general mechanism for existential quantification handling. Let us note that at the time of our work, RDF only had an XML syntax and blank node identifiers did not exist; therefore it was not possible to express some RDF graphs with cycles involving blank nodes. Our proposal was a solution to this problem. To extend RDFS with existential quantification, we introduced the class `:Variable` and the property `:parameter`. A variable is a resource of type `:Variable`; it is linked by a `:parameter` property to the context it belongs to; an existential quantification is represented by an anonymous resource described by a `:referent` property whose value is an instance of class `:Variable`. The scope of a variable is the context it belongs to, just like in first-order logic, where the scope of a variable is the formula it belongs to. As a result, in an RDF graph, an anonymous resource can be duplicated into several anonymous resources coreferencing a same variable; the resulting graph remains semantically equivalent to the initial graph. Figure 3.2 is an example of such an equivalence.

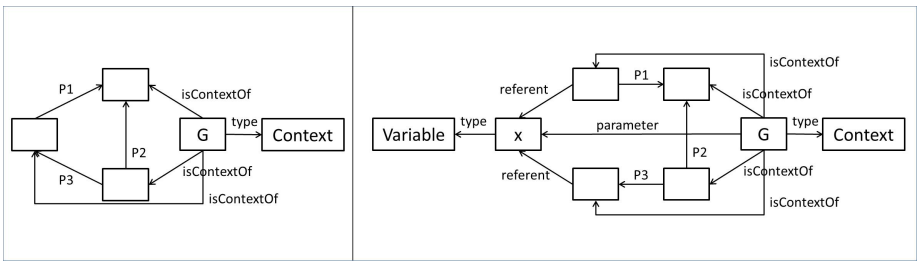

Figure 3.2: Example of two equivalent RDF graphs. On the right, an existential quantification avoids a cycle by considering four anonymous resources instead of three, with two of them related to the same variable

Our proposal for representing contextual knowledge, if conceptually satisfying — based on the CG model —, was hardly tractable, implying large graphs, with many blank nodes, expensive to process and difficult to read by humans. However it was one of the first proposals to answer the question of representing contextual knowledge in RDF and it contributed to draw the attention of the community on this question.

Reification of Relations between Contexts

A few years later, in 2007, with my colleague Olivier Corby, I considered again scenarios motivating the representation of *user* contexts and relations between them [29]. To answer them, we addressed the problem of hierarchically organizing RDF datasets, based on specialization/generalization relations between RDF graphs representing contexts. This led us to propose an extension of RDF, summarized in the following, based again on the CG model. Our proposal was as follows. When considering RDF graphs in a dataset as contexts (each one named by a URI), the root of the hierarchy contains the common data that is true in any context. The other nodes of the hierarchy represent specific contexts; each one recursively inherits the triples of its ancestors and specializes the data embedded in the root context with concurrent sets of triples. As a result, each RDF graph representing a context is a specialization of the graphs representing more general contexts in the hierarchy. The hierarchy of contexts is represented by the logical implication between the RDF graphs representing contexts. This corresponds to the specialization /generalization relation between conceptual graphs defined in the CG model.

To avoid the duplication of triples in this hierarchy of RDF graphs, each node of the hierarchy stores its own specific RDF triples and inheritance of other RDF triples from its ancestors in the hierarchy is dynamically computed. This corresponds to the *join* specialization operation in the CG model. Our proposal is inspired from the notion of *genus/differentia* in the definition of types in the CG model: a concept type has the graph defining its ancestor as genus and the graph specializing it, and therefore differentiating it from its ancestor, as differentia.

We introduced a transitive and reflexive property `:subStateOf` between URIs denoting RDF graphs, reifying the specialization relation between RDF

graphs. The advantage of defining such a property within the RDF model is that context hierarchies themselves can be described and queried to select the context(s) in which further reasoning can be performed. At the time of this work, the SPARQL query language for RDF was born — it was a W3C candidate recommendation. The notion of *named graphs* in SPARQL enables to deal with a form of context, by enabling to limit the search of solutions to a query to a particular graph of a collection of RDF graphs building up the whole dataset considered. The targeted graph is identified, or named, by a URI in the query. We defined SPARQL design patterns based on named graphs to query hierarchies of RDF graphs representing contexts and we proposed a syntactic extension to SPARQL to lighten the writing of such queries. This is presented in [29]. Olivier Corby implemented our proposal in the Corese semantic search engine presented in Section 3.1.4.

The representation of contexts still is a hot topic in KRR, as evidenced by numerous topics related to context-aware applications in Web- or KR- related conferences or by specific workshops like e.g. the international workshops on Acquisition, Representation and Reasoning about Context with Logic (ARCOE-Logic). The need to represent contextual knowledge still is extensively discussed in the semantic Web community, e.g. to address provenance and trust related topics. The introduction of Named Graphs in RDF 1.1 provides a standard way to handle contextual knowledge in semantic Web based knowledge representations.

3.1.3 Representation of Ontological Knowledge for RDF

In 2001, RDFS allowed for the *declaration* of *atomic* classes and properties. The OWL Ontology Web Language was still to come as a W3C recommendation. However the Community of Description Logics was very active in the definition of a language to represent ontological knowledge on the semantic Web. In this context, with Alexandre Delteil, we first addressed the problem of *defining* RDFS classes and RDF properties and we proposed in [51] an extension of RDF model descending from type definition in the CG model. Then we addressed in [48] the problem of expressing *graph* definitions in Description Logics which were underlying the upcoming OWL language. This was again inspired by the CG model. This section summerizes these two contributions.

Extension of RDFS with Class and Property Definitions

Based on the notion of existentially quantified context, as defined in our extension of RDF described in section 3.1.2, we proposed an extension of RDFS with class and property definitions. Let us note that, in fact, our extension of RDF with contexts described in the previous section, was a side effect, a first step in our work aiming at representing classes and properties. In other words, class and property definition was a special use case asking to express contexts in RDF.

Type Definition in the CG Model. A concept type definition is a monadic abstraction, i.e. a conceptual graph whose one generic concept is its formal parameter. It is noted $t_c(x) \leftrightarrow D(x)$. For instance, the following concept type

definition defines a Web page as a document having HTML for representation system:

```
WebPage(x) <=> [Document:x] -> (hasForReprSystem) -> [System:Html]
```

The formal parameter concept node of graph $D(x)$ is called the head of $D(x)$, its type the genus of t_c, and $D(x)$ the differentia of t_c from its genus [88].

Similarly, a relation type definition is a n-ary abstraction, i.e. a conceptual graph with n generic concepts as formal parameters. It is noted $t_r(x_1, \ldots x_n) \leftrightarrow D(x_1 \ldots x_n)$.

Class and Property Definition in RDFS. We defined a class definition in RDFS as a monadic abstraction, i.e. a context whose one resource of type `:Variable` is considered as formal parameter. A property definition is a diadic abstraction, i.e. a context whose two resources of type `:Variable` are considered as formal parameters. For instance, Figure 3.3 presents the definition of class `:WebPage` in the extended RDFS model. A detailed presentation of our proposal

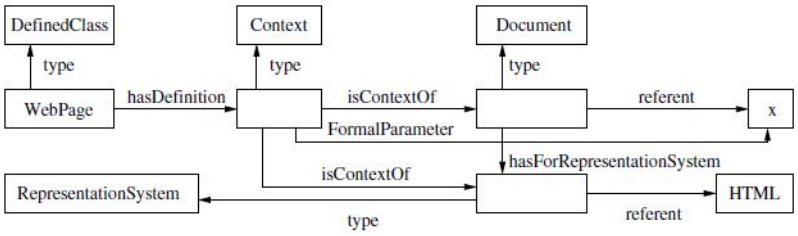

Figure 3.3: Definition of class `WebPage`: a Web page is a document having HTML for representation system.

is available in [51].

A Graph-Based KR Language for Concept Description

The work described in this section took place in the continuation of both an early work of Pascal Coupey and I on the correspondances between Conceptual Graphs and Description Logics [40] and the work described above on the extension of RDF with class and property definition based on the CG model. It has been published in [48].

We proposed a concept description language which combines features of both Conceptual Graphs and Description Logics (DLs). Regarding concept descriptions in the CG model, namely existential, positive and conjunctive graphs, our language is the closure of this language under the Boolean operations. Regarding DLs, it is an extension of $\mathcal{ALC}$ with graph structures in concept descriptions. We called it $\mathcal{GDL}$, standing for Graph Description Logic.

Description Logics distinguish between a terminological language dedicated to the description of concepts and roles (TBox) — corresponding to classes and properties in the RDF model —, and an assertional language dedicated to the statement of facts (ABox). This distinction is similar to the distinction between

the support and the canonical graph base in the CG model. The terminological language of a DL is inductively defined from a set of primitive concepts P_c, a set of primitive roles P_r, the constant concepts $\top$ and $\bot$ and an abstract syntax rules to define concepts and possibly another one to define roles. The subset of constructs used in a given DL determines the expressive power of the latter. The $\mathcal{ALC}$ DL is defined by the following abstract syntax rules:

C_1, C_2	$\longrightarrow$	$\top \mid \bot$	most general | absurd
		$\mid$ P	primitive concept
		$\mid$ $C_1 \sqcup C_2$	concept disjunction
		$\mid$ $C_1 \sqcap C_2$	concept conjunction
		$\mid$ $\neg C_1$	concept negation
		$\mid$ $\forall r.C_1$	universal restriction on roles
		$\mid$ $\exists r.C_1$	existential restriction on roles

$\mathcal{ALC}$ is the basis of more expressive DL. Among them, $\mathcal{GDL}$ is an extension of $\mathcal{ALC}$ to allow graph structures in concept descriptions. Formally, $\mathcal{GDL}$ is inductively defined from P_c, P_r, $\top$ and $\bot$ by the following abstract syntax rule:

C_1, C_2	$\longrightarrow$	$\top \mid \bot \mid P \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \neg C_1$	
		$\mid$ $\forall x G$	existential graph
		$\mid$ $\exists x R$	graph rule

The *existential graph* construct generalizes the *existential restriction* and *universal restriction* constructs of $\mathcal{ALC}$: it enables to introduce graph structures in concept definitions. An existential graph $\lambda x G$ is a concept description consisting of a connected graph G whose concept nodes are all generic, either atomic or defined by a concept description of $\mathcal{GDL}$. One of its concept nodes is designated by the formal parameter x.

The *graph rule* construct is the dual of the *existential graph* construct: the negation of a graph rule concept can be expressed with existential graph concepts and the negation of an existential graph concept with a graph rule concept. This duality is detailed in [47]. A graph rule $\exists x R$ is a concept description consisting of a pair of abstractions $\lambda x x_1 \ldots x_n G, \lambda x x_1 \ldots x_n [C_1 : x_1] \ldots [C_n : x_n]$ where:

- $\lambda x x_1 \ldots x_n G$ is called the hypothesis of the graph rule. It is a connected graph whose $n+1$ concept nodes are designated by the formal parameters $x, x_1, \ldots x_n$. The concepts of G are all generic, either atomic or defined by a concept of $\mathcal{GDL}$.

- $\lambda x x_1 \ldots x_n [C_1 : x_1] \ldots [C_n : x_n]$ is the conclusion of the graph rule. Each C_i is a concept description of $\mathcal{GDL}$. $x, x_1, \ldots x_n$ correspond to coreference links between G and $[C_i : *]_{i=1..n}$, indicating that the generic marker of each C_i represents the same entity as one concept of G.

As a result, $\mathcal{GDL}$ extends $\mathcal{ALC}$ with role intersection, role composition $\circ$, converse of roles $.^-$, and role identity $id(.)$: these constructs can be expressed with graph rule and existential graph constructs as shown in Table 3.1.

A sound and complete tableaux algorithm proving the satisfiability of $\mathcal{GDL}$ in NEXPTIME is provided in [48].

3.1.4 RDF Querying based on the Conceptual Graph Model

Nowadays, the SPARQL recommendation first presents a query language for RDF and then defines entailment regimes to take into account the semantics of

DL constructs	$\mathcal{GLC}$ expressions
$\exists RC$	$[\top : *x_0] \rightarrow (R) \rightarrow [C : *]$
$\forall RC$	$[\top : *x_0] \rightarrow (R) \rightarrow [\top : *x] \Rightarrow [C : *x]$
$\exists (R_1 \sqcap R_2)C$	$[\top : *x_0] \rightarrow (R_1) \rightarrow [C : *x]$ $\searrow (R_2) \nearrow$
$\forall (R_1 \sqcap R_2)C$	$[\top : *x_0] \rightarrow (R_1) \rightarrow [T : *x] \Rightarrow [C : *x]$ $\searrow (R_2) \nearrow$
$\exists (R_1 \circ R_2)C$	$[\top : *x_0] \rightarrow (R_1) \rightarrow [\top : *] \rightarrow (R_2) \rightarrow [C : *]$
$\forall (R_1 \circ R_2)C$	$[\top : *x_0] \rightarrow (R_1) \rightarrow [\top : *] \rightarrow (R_2) \rightarrow [T : *]$ $\Rightarrow [C : *x]$
$\exists (R^-)C$	$[\top : *x_0] \leftarrow (R) \leftarrow [C : *]$
$\forall (R^-)C$	$[\top : *x_0] \leftarrow (R) \leftarrow [\top : *x] \Rightarrow [C : *x]$
$\exists (R \sqcap id(C))D$	$[C \sqcap D : *x_0] \bigcirc (R)$
$\forall (R \sqcap id(C))D$	$[C : *x_0] \bigcirc (R) \Rightarrow [D : *x_0]$

Table 3.1: Expression of $\mathcal{ALC}(\sqcap, \sqcup, \circ, .^-, id(.))$ constructs in $\mathcal{GDL}$

vocabularies while querying RDF data. In 2001, the early age of the semantic Web, researchers from the KRR community addressed the problem of querying RDF as a reasoning task on the whole knowledge representation model, including the ontological knowledge, while researchers from the Relational Databases community focused on implementing efficient algorithm to query RDF data viewed as RDB data. The question of the query language was secondary; a query was viewed as a statement to be proved in the KR model. Moreover, the question of querying assertional RDF data was a secondary reasoning task in the KRR community: the DL community was focusing on ontological reasoning (concept satisfiability, concept classification) to design and maintain high quality ontologies.

The KRR approach clearly appears in our publications on the design of the Corese semantic search engine [27] [24] [25] [28]. We summarize them in this section; we first present our point of view on RDF Querying as an ontology-based reasoning task, then our approach based on the CG model and finally the query language.

RDF Querying as an Ontology-based Reasoning Task

In the design of Corese, we early focused on *ontology-based* querying, taking into account RDFS semantics, while most RDF query engines were only taking into account RDF semantics. The ontological knowledge representation language handled by Corese in 2002 was RDFS extended with meta-properties to express algebraic properties of RDF properties: symmetry, transitivity and reflexivity, and inverse properties [27]. Corese progressively evolved to implement OWL-Lite after the recommendation of the Ontology Web Language (OWL) by W3C.

We envisioned ontology-based querying of RDF KB according to a logical model of information retrieval [24]: Given (1) a model for ontologies (RDFS), (2) a model for annotations of resources based on ontologies (RDF), (3) a model for queries based on ontologies, and (4) a matching function defining how a query is matched with any annotation, a Web resource R is relevant for a query

Q according to the ontology O from which they are built *iff* the annotation of R and the ontology O together logically imply Q. The query is viewed as a set of constraints on the description of the Web resources to be retrieved and then corresponds to a search problem to be solved. The matching function implements the strategy chosen for solving this problem. This is exactly what we now call *entailment regimes* in the SPARQL recommendation.

We presented Corese as an ontology-based semantic search engine for the semantic Web that implements such a matching function using the projection operator defined in the CG model. When matching a query with an RDF annotation, according to a shared RDFS ontology, these were translated in the CG model. This translation was based on the correspondences between RDF and CG models described in Section 3.1.1. Basically, both the RDF graph to be queried and the query were translated into conceptual graphs, and the class hierarchy and property hierarchy of the RDFS vocabulary were translated into the CG support. Then the solutions to the query were computed by using the projection operator of the CG model to compute the solutions of the query. Let us note that RDFS vocabularies were also translated into conceptual graphs, like any piece of RDF data, which enabled to query the ontological knowledge as well as the assertional knowledge. This was an original feature of Corese.

Based on this RDF2CG translation, we took advantage, for the design of Corese, of previous works in this KRR community, in particular contributions on operators and reasoning capabilities in the CG formalism.

RDF Querying based on the CG *Projection* Operator

The projection operation is the basis for reasoning in the CG model. A conceptual graph G_1 logically implies a conceptual graph G_2 iff it is a specialization of G_2 (noted $G_1 \leq G_2$), and G_1 is a specialization of G_2 iff there exists a projection of G_2 into G_1 such that each concept or relation node of G_2 is projected on a node of G_1 whose type is the same as the type of the corresponding node of G_2 or a specialization of it, according to the concept type hierarchy and the relation type hierarchy.

Formally, let us define a CG as a labeled bipartite graph $G = (C, R, E, l)$ where C and R are the sets of its concept nodes and of its relation nodes, E is the set of its edges and l is a mapping which labels each relation node r of R by a relation type *type(r)* of the relation type hierarchy $\mathcal{T}_r$ and each concept node c of C by a couple *(type(c), ref(c))* where *type(c)* is a concept type of the concept type hierarchy $\mathcal{T}_c$ and *ref(c)* is an individual marker or the generic referent $*$. The projection operation is then defined as follows [22]: A projection from a CG $G = (C_G, R_G, E_G, l_G)$ to a CG $H = (C_H, R_H, E_H, l_H)$ is a mapping Π from C_G to C_H and from R_G to R_H which:
- preserves adjacency and order on edges: $\forall rc \in E_G$, $\Pi(r)\Pi(c) \in E_H$ and if c is the i^{th} neighbor of r in G then $\Pi(c)$ is the i^{th} neighbor of $\Pi(r)$ in H;
- may decrease labels: $\forall x \in C_G \cup R_G, l_H(\Pi(x)) \leq l_G(x)$.

A query was processed in the Corese engine by projecting the corresponding conceptual graph into the conceptual graph translated from RDF. The retrieved resources and literal values are those for which there exists a projection of the query graph into the target graphs. Corese initially output as solutions RDF graphs built from the existing projections, in the RDF/XML syntax. It evolved during times to implement the recommended XML SPARQL result.

The matching algorithm of Corese is described in [28]. We proposed an efficient algorithm relying on two main principles. First, the search space for node projections is limited by dealing with relations first and ordering them so as to force node projections. Second, our algorithm integrates the evaluation of value constraints during the search for graph homomorphisms to efficiently reduce the search space.

Resolving Value Constraints *while* Pattern Matching. A sequential algorithm where the resolution of value constraints would succeed pattern matching would be quite inefficient. Our algorithm takes into account value constraints during the search for a graph projection: while searching for a projection of a query graph Q into a graph G, as soon as a value in G is rejected because it does not satisfy a value constraint, the projection as a whole which involves this value can be rejected. Moreover, the sooner value constraints are taken into account the smaller the search space becomes. Therefore our algorithm handles value constraints as soon as they are evaluable.

Highest Precedence for Relations in Conceptual Graphs. Our algorithm takes advantage of the hypergraph structure of our representation of conceptual graphs to limit the search space for node projections. We represent a conceptual graph by a hypergraph whose hyperarcs are the relation nodes and the adjacent concept nodes of each one (nodes are those of the conceptual graph). As a result, when searching for homomorphisms, relations are no longer nodes: they are viewed as constraints for (concept) node projection. Nodes are no longer projected in isolation but each one is projected at the same time as the other arguments of a chosen relation to which it participates; relations thus are constraints which reduce the search space of possible projections of nodes. This principle is close to the one described in [41]. Formally, let U the set of hyperarcs or *relations* of G that can be defined as tuples of nodes adjacent to the same relation node. we choose a first relation $r = (x_1, \ldots x_i) \in U(Q)$, such that $\forall t \in type(r), \exists r' = (x'_1, \ldots x'_i) \in U(G)$ such that $\exists t' \in type(r')$ with $t' \leq t$. This choice determines the projections $\pi(x_1) = x'_1, \ldots \pi(x_i) = x'_i$ of $x_1, \ldots x_i$. While doing so the theoretical search space for the projection of r, $C_G \times \ldots \times C_G$, becomes the extension of t'. Moreover, when dealing with the next chosen relations, some of their arguments will already have projections previously chosen and the search space for the remaining arguments will even more decrease.

Finally, the keystone of an efficient implementation of the projection operation in Corese was the compilation of the CG support into compiled type hierarchies, a compiled type being associated to each resource in the query and target conceptual graphs. The key features of the matching algorithm of Corese were three:

Ordering Relations in the Query Graph. Relations in the query graph Q are heuristically ordered to constrain at best the search space. Heuristics are based on both the structure of query graph Q and the RDF graph G. Regarding the query graph structure, the ordering depends on both the connexity of relations on their arguments and the occurrence of value constraints associated to relation arguments. By choosing a relation connected by the greatest num-

ber of arguments to previously chosen relations of Q, these arguments already have projections which diminish the search space for the remaining arguments. Furthermore, the more value constraints on nodes are evaluated, the more the search space will diminish. At each step of the search we choose to handle the relation for which the greatest number of constraints are evaluable. Regarding graph G, the ordering depended on the types of relation types in G and the number of their occurrences. The early choice of the relations whose type occur the least in G significantly reduces the search space.

Graph Indexing and Candidate Relations. Graph G is indexed by relation types and by each argument of the relations. Hence there is a direct access to the list of relations of a given type involving a given node. This graph indexing is a preliminary step of the matching algorithm; it is preprocessed and statically stored. Based on this static index of G, we associate to each relation $r \in U(Q)$ a set $candidates(r)$ of relations of $U(G)$ candidates for arguments of r to be projected on theirs: $candidate(r) = \{s \in U(G), type(s) \leq type(r)\}$. The backbone of the matching algorithm is the stack of the ordered relations of $U(P)$ associated to their candidate lists. Candidate lists initially correspond to the static index of G;their sizes incrementally are reduced as they are piled up according to the heuristic criteria described above. Their decreasing is as follows. Let r the current relation elected to be piled up. If it is connected to some relation r' previously piled up with the i^{th} argument of r being the j^{th} argument of r', then relations in $candidate(r)$ can be eliminated whose i^{th} argument does not appear as j^{th} argument in $candidate(r')$. Moreover, if some value constraint is evaluable once r is piled up, $candidate(r)$ is further decreased by eliminating candidates for which the constraint evaluates to false. As a result, let $stack(Q)$ the stack where all relations of $U(Q)$ are piled up; it constitutes the search space for the graph projection search.

Backjump. Our algorithm incrementally searches for a partial projection for nested subgraphs of Q. To build these subgraphs, relations are considered according to their ordering in $stack(Q)$. This static ordering enables the handling of constraints during the projection search without ever and ever testing their evaluable status at each step of the algorithm, which would be too time consuming. Based on this static ordering of relations defined by $stack(Q)$, in case of failure of a partial projection search, our algorithm does not just systematically backtrack to the preceding relation in the stack but possibly goes to a deeper relation. It directly backjumps to the relation which solves the failure: the latest relation which binds (for the first time) one of the variables in the failing relation or the failing filter.

A Query Language for RDF descending from the CG Model

The query language of Corese basically was RDF with some additional features to represent variables (a variable name is prefixed with a question mark) and constraints on property values and on types. Here again, we can recognize the influence of our background in the CG model where a concept node in a graph is a couple of a type and a marker. An 'RDF' query was interpreted as a (query) conceptual graph and processing instructions for the projection function

(constraints to verify and subgraphs for which the existence of a projection is optional).

The Corese query language comprised most features of the upcoming SPARQL query language. It slightly evolved during time and once the SPARQL query language for RDF was proposed as a recommendation, it was compliant with it (it was a superset of it). Let us note that the Corese query language early enabled to construct queries with variables at the place of properties which was rarely encountered in the first implementations of RDF query engines. This came again from our "conceptual graph background" where properties are nodes, handled in a similar way to concept nodes. Even more notable, the Corese query language early provided means to express property paths which will be introduced in the working draft of SPARQL 1.1 in 2010[2]. At that time, we envisioned it as a form of approximation, property paths in query graphs enabling to search for resources connected to each other through a possibly unknown number of unknown properties).

The syntax of the Corese query language also evolved during time, starting from a RDF/XML-like syntax in its first version [27], to end-up with a SPARQL-like syntax in [25], with triples representing either properties between resources or constraints on property values or types. Finally, the SPARQL syntax and semantics was adopted in [28]

Extension of the CG *Projection* Operator for Approximate Search

We early addressed the problem of *approximation* to answer user-centered scenarios and corollary research questions further discussed in the next chapters. To answer them, we integrated the notion of approximate search in Corese. We distinguished between structural approximation based on paths of unknown properties in the query graph and ontology-based approximation based on semantic distances between classes or between properties in the ontology. We integrated the results of the PhD thesis of Fabien Gandon on semantic distances [61] and extended the projection operation of the CG model to handle such approximations. This is described in [24] and [25]. In short, we extended the definition of the CG projection as follows in order to allow the query graph to be projected with concept and relation types not necessarily subsumed by those of the query but semantically close enough to them in the ontology.

Definition 1 *An approximate projection from a CG* $G = (C_G, R_G, E_G, l_G)$ *to a CG* $H = (C_H, R_H, E_H, l_H)$ *is a mapping* Π *from* C_G *to* C_H *and from* R_G *to* R_H *which:*
- preserves adjacency and order on edges: $\forall rc \in E_G$, $\Pi(r)\Pi(c) \in E_H$ *and if* c *is the* i^{th} *neighbour of* r *in* G *then* $\Pi(c)$ *is the* i^{th} *neighbour of* $\Pi(r)$ *in* H*;*
- may change the labels of concept nodes to ontologically close ones:
$\forall c \in C_G, D_{\mathcal{T}_c}(type(c), type(\Pi(c))) < \varepsilon$*, where* $D_{\mathcal{T}_c}$ *is the ontological distance in the concept type hierarchy and* ε *is a threshold chosen as the maximal distance allowed.*
- may decrease the labels of relation nodes or change them to semantically close ones: $\forall r \in R_G, l_G(r) \leq l_H(\Pi(r))$ *or a* rdf:seeAlso *property stands between* $l_G(r)$ *and* $l_H(\Pi(r))$

[2] https://www.w3.org/TR/2010/WD-sparql11-query-20100601/

We extended the definition of the CG projection as follows in order to allow the mapping of a relation node with a graph path.

Definition 2 *Let $\mathcal{P}_G$ the set of the relation path graphs in a CG G. A projection from a CG $G = (C_G, R_G, E_G, l_G)$ to a CG $H = (C_H, R_H, E_H, l_H)$ is a mapping Π from C_G to C_H and from R_G to $\mathcal{P}_H$ which:*
- preserves adjacency and order on edges:

- $\forall rc \in E_G$, *if* $\Pi(r) \in R_H$ *then* $\Pi(r)\Pi(c) \in E_H$ *and if c is the* i^{th} *neighbour of r in G then* $\Pi(c)$ *is the* i^{th} *neighbour of* $\Pi(r)$ *in H;*
- $\forall rc \in E_G$, *if* $\Pi(r) \notin R_H$ *then* $\Pi(r)$ *is a path graph defining a relation type t and considering the contraction in H of* $\Pi(r)$ *to a relation node* r' *of type t,* $r'\Pi(c) \in E_H$ *and if c is the* i^{th} *neighbour of r in G then* $\Pi(c)$ *is the* i^{th} *neighbour of* r' *in H;*

- may decrease labels:

- $\forall c \in C_G$, $l_G(c) \leq l_H(\Pi(c))$;
- $\forall r \in R_G$, *if* $\Pi(r) \in R_H$ *then* $l_G(r) \leq l_H(\Pi(r))$;
- $\forall r \in R_G$, *if* $\Pi(r) \notin R_H$ *then* $\forall x \in R_{\Pi(r)}$, $l_{\Pi(r)}(x)) \leq l_G(r)$

The combination of both ontology-based and structural approximations combines the above two definitions.

3.2 A Knowledge Graph Model for the Semantic Web

We gradually freed ourself from the specific Conceptual Graph model over time, going after a Knowledge Graph model that would better fit with the semantic Web standards. Our perspective thus evolved to the question of *Which generic graph-based model does enable KRR on the semantic Web?*.

During the years of Corese development we developed relationships with the French community working on Conceptual Graphs, especially the RCR team which has become the GraphIK led by Marie-Laure Mugnier at LIRMM. In 2005, I co-organized with Olivier Corby and two other colleagues a workshop on Reasoning the semantic Web with Graphs at the French Artificial Intelligence platform [68]. In 2008, I participated to the GRIWES project led by Fabien Gandon on the development of a generic Knowledge Graph model for the semantic Web. This is described in Section 3.2.1. One of the motivations behind this project was the interoperability of graph-based KKR tools developed by several teams, in particular of Cogitant developed by RCR and dedicated to conceptual graph reasoning and Corese developed by Edelweiss dedicated to a Conceptual Graph operationalisation of RDF/S. This final goal has never been achieved. However this has been the starting point for the re-conception re-implementation of Corese into KGRAM. This is described in section 3.2.2. The architecture of KGRAM enabled us to answer new scenarios of the semantic Web among which the integration of heterogeneous and distributed data. This is described in section 3.2.3.

3.2.1 GRIWES: Graph-based Representations and Inferences for Web Semantics

This section summarizes the results of the GRIWES project, published in [1]. In order to develop a common model and an open-source freeware platform to share state-of-the-art structures and algorithms across several specific graph-based knowledge representation frameworks, we defined a general framework distinguishing three layers of abstraction and one transversal component for interaction:

- the *Structure layer* gathers and defines the basic mathematical structures (e.g. oriented acyclic labelled graph) that are used to characterize the primitives for knowledge representation (e.g. type hierarchy);
- the *Knowledge layer* factorizes recurrent knowledge representation primitives (e.g. a rule) that can be shared across specific knowledge representation languages (e.g. RDF/S, Conceptual Graphs);
- the *Language and Strategy* is two-sided; one side gathers definitions specific to languages (e.g. RDF triple) and the other side identifies the strategies that can be applied to these languages (e.g. validation of a knowledge base, completion of a fact by rules);
- the *interaction and interfaces* aspect was deemed transversal to the above layers. It gathers events (e.g. additional knowledge needed) and reporting capabilities (e.g. validity warning) needed to synchronize conceptual representations and interface representations.

In the framework of our one-year funded project, we focused on and achieved the definition of the Structure and Knowledge layers described in the following.

The GRIWES Structure Layer

The structure layer is the core layer of the architecture of GRIWES. It gathers the basic mathematical structures that we chose to characterize the primitives for KRR, namely the notion of *entity-relation graph* as the keystone for higher-level graph-based representation, the notions of *mapping* between two such graphs and *proof of a mapping* reifying a mapping as the keystones for reasoning on entity-relation-graphs, and the notion of *constraint system* both for knowledge representation and efficient implementation purposes. These are described in the following.

An Entity-Relation graph (ERGraph) is intended to describe a set of entities and relationships between these entities, an entity being anything that can be the topic of a conceptual representation. A relationship might represent a property of an entity or relate two or more entities. It can have any number of arguments including zero and these arguments are totally ordered. In graph theoretical terms, an ERGraph is an *oriented hypergraph*, where nodes represent the entities and hyperarcs represent the relations on these entities. However, a hypergraph has a natural graph representation associated with it: a bipartite graph, with two kinds of nodes respectively representing entities and relations, and edges linking a relation node to the entity nodes arguments of the relation; the edges incident to a relation node are totally ordered according to the order

on the arguments in the relation. The nodes (entities) and hyperarcs (relations) in an ERGraph have labels. At the structure level, they are just elements of a set L that can be defined in intention or in extension. Labels obtain a meaning at the knowledge level.

In some knowledge representation primitives and some algorithms it is useful to distinguish some entities of a graph. For this purpose we defined a second core primitive, called λ-ERGraph. A *λ-ERGraph* λG is a couple of an ERGraph G and a tuple of entities distinguished in G.

Mapping entities of graphs is a fundamental operation for comparing and reasoning with ERGraphs. It is a basic operation used in many more complex operations e.g. rule application. An *EMapping* from an ERGraph H to an ERGraph G is a partial function M that associates each entity of H with at most one entity of G. By default an EMapping is partial. This enables to manipulate and reason on EMappings during the process of mapping graphs. When this process is finished, the EMapping — if any — is said total: all the entities of H are mapped with an entity of G.

In general specific mappings are used that preserve some chosen characteristics of the graphs (e.g., compatibility of labels, structural information etc.). In particular we defined an ERMapping which constrains the structure of the graphs being mapped and an EMapping$_{<X>}$ which constrains the labelling of entities in the graphs being mapped. An ERMapping is an EMapping that leads to map each relation in H to a relation in G with the same arity. An EMapping$_{<X>}$ is an EMapping that satisfies a compatibility relation X on entity labels. An ERMapping$_{<X>}$ is both an ERMapping and an EMapping$_{<X>}$. A Homomorphism is a total ERMapping. The notion of projection as defined in Conceptual Graphs corresponds to a Homomorphism$_{<X>}$ that is to say a total ERMapping$_{<X>}$, where X is a preorder over the label set L.

A proof of a mapping as a kind of "reification" of the mapping: it provides a static view over the dynamic operation of mapping, enabling thus to access information relative to the state of the mapping. Formally the *EProof* of an EMapping from H to G is the set(s) of associations detailing the exact association from each entity and relation of the query graph H to entities and relations of G. We associate with each kind of EMapping a kind of proof: EProof, ERProof, EProof$_{<X>}$ and ERProof$_{<X>}$.

An EMapping constraint system is a function $\mathcal{C}$ that sets additional conditions that an EMapping must satisfy in order to be correct. It takes the form of an evaluable expression which must evaluate to *true* for an EMapping to satisfy the constraint system. We introduced this notion (1) to represent SPARQL FILTER clauses and equivalent features in other graph-based query language, and (2) to provide efficient access means to indexes of graphs, for instance to retrieve all the arcs of a graph satisfying a given constraint system.

The GRIWES Knowledge Layer

In the GRIWES architecture, a knowledge base comprises a *vocabulary*, one or several bases of *facts*, optionally a base of *rules* and a base of *queries*. These elements are defined based on the notions defined in the structure layer.

A vocabulary is a set of non necessarily disjoint sets, with preorders. A fact is an ERGraph which entity and relation labels are elements of the vocabulary of the knowledge base.

A query is a couple of a λ-ERGraph and a Constraint system C. The answers to a query depend on the type X of EMapping used to query the base. An X-*Answer* to a query Q in a fact F is such that there exists an EMapping M from the ERGraph G of Q to F such that M maps all the entities distinguished in G to elements of A. The proof of an X-Answer is the proof of the EMapping associated to that X-Answer.

A rule $R = (H, C)$ is a couple of a query $H = (G, C)$ and a λ-ERGraph C of the same size as G. H is the hypothesis of the rule, and C is its conclusion. R is X-applicable to a fact F iff there exists an X-Answer to H in F. The X-Application of R on F with respect to A merges C in (A, F).

Additional notions are introduced in the knowledge level to represent the abstract operations that can be performed on a knowledge base.

An ERFunction F is a function associating to an ERProof P a label or an error. A functional ERGraph is an ERGraph where some entities or relations are labelled with ERFunctions. The evaluation of a functional ERGraph G with respect to an EProof P and an environment E is a copy G' of G where every functional label is replaced by the evaluation of the function against P. If any of the evaluations returns an error then $G' = \emptyset$. A functional rule is a rule whose conclusion is a functional λ-ERGraph. Let $R = (H, C)$ be a functional rule X-applicable to a fact F, and A be an X-Answer to H in F and P be a proof of that X-Answer. The X-functional-Application of R on F with respect to P merges the evaluation of C with respect to P in (A, F).

The normal form $NF(G)$ of an ERGraph G is the ERGraph obtained by merging every entities of a same equivalence class defined by its co-reference equivalence relation R over its set of entities into a new entity calculated by calling a $fusion$ function on the entities of this class. Co-reference and fusion are abstract functions which must be specified at the language level.

We validated the GRIWES abstract pivot model by representing both the semantics of the RDF language and the semantics of the Conceptual Graph language.

As mentioned in the conclusions of [1], there remained several open questions on our GRIWES model at the end of this one-year project and no implementation had been conducted. We did not manage to get the continuation of the project funded by the ANR agency; after two unsuccessful submissions, the collaboration between our teams ended. But within the Edelweiss team this has marked a key turning point towards the re-conception of Corese, more generic, freed from the specific model of Conceptual Graphs, relying on an abstract knowledge graph model close to that of GRIWES.

3.2.2 KGRAM: Knowledge Graph Abstract Machine

In the continuation of the abstraction work we conducted in GRIWES, Olivier Corby and I designed the KGRAM semantic Web engine (acronym for Knowledge Graph Abstract Machine): we identified high level abstract primitives descending from the GRIWES model which constitute both KGRAM's query language and API. This section summarizes this work which has been published in [30] and [31]; we first present the abstract query language of KGRAM, then the interpreter of KGRAM query language and finally KGRAM API.

KGRAM Abstract Query Language

The abstract syntax of KGRAM query language is given by the following grammar:

```
QUERY  ::= query(NODE *, EXP)
EXP    ::= QUERY | NODE | EDGE | FILTER | PATH
         | and(EXP, EXP) | union(EXP, EXP) | option(EXP) | not(EXP)
         | exist(EXP) | graph(NODE, EXP)
NODE   ::= node(label)
EDGE   ::= edge(label, NODE *)
PATH   ::= path(RegExp, NODE, NODE)
FILTER ::= filter(FilterExp)
```

A QUERY expression enables to represent a query. Its parameter EXP represents the expression to be evaluated and its parameters NODE the variables for which the list of values is searched when the expression is evaluated on the graph which is queried. A QUERY expression also enables to formulate a query nested into another.

NODE and EDGE expressions enable to query for nodes or n-ary relations. A FILTER expression enables to filter the retrieved nodes or relations: its `FilterExp` parameter is a boolean expression of a constraint language enabling to express constraints on the searched nodes or relations in the graph which is queried:

```
FilterExp ::= Variable | Constant | Term
Term      ::= Oper(FilterExp *)
Oper      ::= '<' | '<=' | '>=' | '=' | '!='
            | '&' | '|' | '!' | '+' | '-' | '*' | '/' | FunctionName
```

Let us note that NODE, EDGE and FILTER expressions are primitive and we will show at the end of this section that they correspond to interfaces of the abstract machine KGRAM.

A PATH expression is a generalization of an EDGE expression. It enables to query for paths of binary relations between two nodes in a graph. Its `RegExp` parameter is a regular expression describing a set of relation paths:

```
RegExp ::= label | RegExp '*' | RegExp'/'RegExp | RegExp'|'RegExp
```

FOr instance, here is an example of a query with a PATH expression enabling to retrieve all the elements in a list:

```
query({node('?y')}, path(rdf:rest*/rdf:first, node('?x'), node('?y')))
```

A `and()` (resp. `union()`) expression enables to express a conjunction (resp. disjunction) between two expressions. An `option()` expression makes optional the existence of solutions to some expression in the search of solutions to a query. A `not()` expression expresses negation as failure. An `exist()` expression enables to search for only one solution (the first retrieved). A `graph()` expression enables to specify the knowledge graph upon which the query is evaluated.

Depending on the subset of query expressions that we consider, we define a particular (sub) language for KGRAM. Worth noticing, the NODE and EDGE expressions define a query language corresponding to the one of the Simple Conceptual Graph model [87, 21]. By including FILTER expressions, we consider conceptual graphs with constraints [2]. The expressions NODE, EDGE, FILTER, `and()`, `union()`, `option()` and `graph()` define a sub-language which corresponds

$$\frac{match(ENV \vdash NODE \rightarrow LENV) \wedge merge(ENV, LENV \rightarrow LENV')}{ENV \vdash NODE \rightarrow LENV'} \qquad \frac{match(ENV \vdash EDGE \rightarrow LENV) \wedge merge(ENV, LENV \rightarrow LENV')}{ENV \vdash EDGE \rightarrow LENV'}$$

$$\frac{eval(ENV \vdash F : false)}{ENV \vdash filter(F) \rightarrow nil} \qquad \frac{eval(ENV \vdash F : true)}{ENV \vdash filter(F) \rightarrow list\ ENV}$$

Table 3.2: Rules of Natural Semantics to evaluate the NODE, EDGE and FILTER expressions of KGRAM's query language: the top-left rule for NODE expressions, the top-right rule for EDGE expressions and the two bottom rules for FILTER expressions

to the core of SPARQL SELECT query form extended to n-ary relations. In addition, the `exist()` expression corresponds to the ASK query form. The notion of nested query captured in the `query()` expression and that of relation path captured in the `path()` expression are in SPARQL 1.1.

Interpreter of KGRAM Query Language

Natural Semantics. We specified the semantics of the interpreter of KGRAM query language in Natural Semantics [67]. This formalism was initially developed for programming languages, with axioms and inference rules characterizing each language construct. An inference rule is applied within an environment and produces one or several new environments. In the case of KGRAM, an environment represents a set of bindings of query variables with values. This corresponds in the GRIWES knowledge layer to an ERProof$_{<X>}$. The rules we established for KGRAM's query language describe the evolution of the environment (initially empty) during the evaluation of an expression building up a query.

The rules to evaluate NODE, EDGE and FILTER expressions are presented in Table 3.2. The evaluation of a NODE or EDGE expression in an environment *ENV* requires to compute the list of environments *LENV* capturing the possible matching of NODE or EDGE in the graph which is queried and to merge *ENV* and *LENV*. These two operations are synthesized in the rule bases *match* and *merge* which specify the semantics of respectively the comparator of node or edge labels and the environment manager of KGRAM.

In the two rules to evaluate a FILTER expression, the rule base *eval* is relative to the evaluation of the boolean expression by which a FILTER expression is parametrised; it exploits the bindings of the query variables embedded in the current environment *ENV*. The rules specify that if this boolean expression is evaluated to false then an empty environment list (nil) is produced (there is no solution), otherwise the list produced contains a single element which is the current environment (this list is created with the *list* operator).

In the same way, we have defined similar rules for each other expression of KGRAM's query language that we do not present here. The complete set of Natural Semantics rules of the language defining the way its expressions must be evaluated are presented in [31].

Evaluation Function. The core of KGRAM is its evaluation function which interprets KGRAM's abstract query language. Its algorithm implements the rules of Natural Semantics specifying the semantics of the language. It specially relies on the above described rules associated to expressions NODE, EDGE and FILTER. The operationalisation of the rules associated to expressions NODE and EDGE corresponds to the search of homomorphisms on labelled graphs whose relations may be n-ary. The operationalisation of the rules associated to expression FILTER corresponds to the search of homomorphisms with constraints.

Listing 3.1 shows KGRAM's core algorithm. The `queryStack` argument of the `eval` function represents the stack of expressions participating to the query that is evaluated. Its argument `i` represents the current position in this stack. The function is initially called with the whole query in the stack and a value of zero for i. An instance of KGRAM is created with (1) a `producer` responsible for the production of candidate nodes and edges of the data graph matching those of the query graph, (2) a `matcher` responsible for the matching of query and target nodes or edges, (3) a `filterer` responsible for the evaluation of constraints (filters), (4) an environment manager `env` responsible for the storage in a stack structure of the current environment, i.e. a partial homomorphism described as node bindings and (5) a list of complete homomorphisms (representing the results of the evaluated query expression). We will see in the following that the `producer`, the `matcher` and the `evaluator` called in this algorithm both implement KGRAM's application programming interfaces. This is what ensures the independence of the interpreter of the query language from the data models which can be queried and therefore the interoperability of KGRAM.

```
eval(queryStack, i){
  if (queryStack.size() = i) {store(env); return;}
  exp = queryStack(i);
  switch(exp){
    case EDGE:
      for (Edge r : producer.getEdges(exp, env)){
          if matcher.match(exp, r){
            env.push(exp, r);
            eval(queryStack, i+1);
            env.pop(exp, r);}}
      break;
    case NODE: // similar to case EDGE
    case FILTER: if (filterer.test(exp, env)) eval(queryStack, i+1);
  }}
```

Listing 3.1: KGRAM's core algorithm

KGRAM Application Programming Interface

Abstract Data Structures. The KGRAM interpreter accesses the queried data bases through an abstract API that hides the graph's structure and implementation: it operates on a graph abstraction by means of abstract structures and functions and it ignores the internal structure of the nodes and edges it manipulates to evaluate a query expression over a target graph. More precisely, the target graph is accessed by node and edge iterators that implement the *Node* and *Edge* interfaces of KGRAM. These are the very same interfaces that operationalize the NODE and EDGE expressions of KGRAM's query language.

As a result, KGRAM can process any kind of knowledge graph, among which conceptual graphs (with n-ary relations) as well as RDF graphs (with binary relations). It remains independent of any graph implementation and any data structure.

Abstract Operators. KGRAM accesses the target graph through an abstract graph manager which implements its *Producer* interface. This graph manager enumerates the graph nodes and edges (implementing the *Node* and *Edge* APIs) that match the nodes and edges occurring in a given expression (and implementing the same APIs). It uses the KGRAM APIs of a node and edge matcher described bellow and thus ignores the way nodes or edges are matched.

A node and edge matcher implements the KGRAM *Matcher* interface. It is responsible for comparing node and edge labels. It implements the *match* semantic rule base occurring in the rules of natural semantics specifying the expressions NODE and EDGE of the query language. Depending on the *Matcher* implementation, the label comparison consists in testing string label equality or it may take into account class and property subsumption, or compute approximate matching based on semantic similarities, etc.

Constraints are abstract entities that implement the *Filter* interface which specify the FILTER expression of the query language. Filters are evaluated by an object that implements the *Evaluator* interface. KGRAM ignores the internal structure of filters, it calls the `eval` function of *Evaluator* on *Filter* objects and passes the *Environment* as argument. This *eval* function implements the *eval* rule base occurring in the rules of Natural Semantics 3.2 of FILTER.

Figure 3.4 presents the resulting architecture of KGRAM query engine.

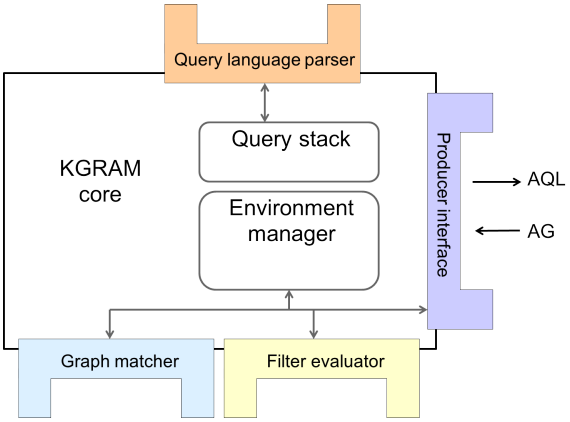

Figure 3.4: KGRAM core query engine

Interoperability. KGRAM comes with a default implementation of its APIs, for querying RDF with SPARQL 1.1 with RDFS entailment[3]. It is developed and maintained by Olivier Corby. We showed that our goals of abstraction and interoperability with other graph-based reasoning framework were achieved

[3] `http://wimmics.inria.fr`

through the development of two additional implementations of KGRAM APIs with both Corese and Jena[4] [69]. The connection to Corese was almost immediate because KGRAM was designed as an abstraction of the principles of Corese. In this port, KGRAM handles its whole query language, and queries RDF graphs implemented as conceptual graphs. We have also ported KGRAM (except property path) on Jena (we co-supervised the Master trainee of Corentin Follenfant on this subject). This required less than 1000 lines of code.

3.2.3 Querying Heterogeneous and Distributed RDF Data with KGRAM

Nowadays, a growing number of applications expect to manipulate the Linked Data seamlessly available over the Web. Conversely, Linked Data opens new opportunities to enrich existing applications by exploiting enlarged, joined data and knowledge repositories. This momentum creates the need for new tools able to query, join, and manipulate the heterogeneous and distributed data sources composing the Web of Linked Data. The GRIWES model and KGRAM's design principles have broaden the perspective to answer these needs. This section shows how KGRAM supports (i) query-based federation of multiple data sources; (ii) mediation of a wide range of heterogeneous data models encountered on the Web; (iii) access to distributed data sources. These are the results of a collaborative work with Olivier Corby, Johan Montagnat and Alban Gaignard published in [39], [60] and [59]. Olivier Corby and I participated to the supervision of the PhD thesis of Alban Gaignard [58] directed by Johan Montagnat.

Federation of Multiple Data Sources

KGRAM query evaluation algorithm makes it easy to query multiple data sources through multiple data producers. A meta-producer, interfaced to the query engine of KGRAM on the one side and to multiple data producers on the other side, just forwards the graph node and edge queries of KGRAM core query engine to all attached producers before merging all resulting bindings received into a single environment returned to the query engine.

Integration of Heterogenous Data

The implementation of KGRAM's `Producer` interface by a metaproducer is also the key to integrate data with heterogeneous knowledge models. It only requires that an implementation of the `Producer` interface is implemented for each knowledge model, with also different implementations of the `Node` and `Edge` interface, and that a metaproducer iterates over all of them. In that case, the matcher of the query engine which is called by the interpreter for each candidate node or edge returned by the metaproducer is here to harmonize the semantics of the preliminary matchings of the producers.

[4] `http://jena.sourceforge.net/`

Data Distribution

The implementation of the data producer and its capability to return only the necessary graph components to the query engine is critical regarding the performance of KGRAM, especially in a distributed environment were data sources are remote and the graph components are communicated over the network. In a naive implementation, a remote producer can passively deliver all graph nodes and edges corresponding to its entire data base for matching and filtering by the (remote) query engine. However, this strategy is fairly inefficient in case a large number of the graph components sent to the query engine are later on discarded by the matcher or the filter evaluator. Consequently, the interface to the producer also includes parameters to transfer the known bindings and the filters to it. This allows for the implementation of advanced operations such as source data filtering and partial matching inside data producers.

Concurrent querying of multiple data sources is an obvious optimization when querying multiple data sources, especially if these are remotely located and each source query is processed by different computing units. A parallel meta-producer that implements a meta-producer connected to data sources queried simultaneously rather than sequentially was added to the KGRAM software suite.

Implementations of the Producer API

Several base *Producer* implementations are included in the KGRAM software suite (Figure 3.5a): for RDF datasets in raw RDF/XML files, N3 text files or Jena databases, and for mediated XML data sources. The bottom row of Figure 3.5 also shows three other specific *Producer* implementations available. The Meta-Producer (3.5b) enables the connection to multiple data sources. It exposes a *Producer* interface and requires multiple other *Producer* components to be connected through the same interface. It simply forwards input queries to the connected *Producers* and merges all results delivered by all subsequent *Producers* before delivering them to the query engine. The Producer client (3.5c) is an interface to any SPARQL remote endpoint. It transforms inbound queries in the abstract query language of KGRAM into SPARQL queries that are sent over HTTP to any SPARQL-compliant endpoint. The entities returned by the endpoint are then transformed into abstract knowledge graph results. Since the abstract query language of KGRAM extends SPARQL 1.1, KGRAM can connect to any SPARQL endpoint. KGRAM software suite also contains a KGRAM-enabled endpoint as illustrated in Figure (3.5d). The producer client communicates queries in KGRAM abstract query language in a concrete format, rather than raw SPARQL, to the KGRAM endpoint. The endpoint uses a local KGRAM instance to parse and evaluate these queries.

Applications

To illustrate the versatility of KGRAM architecture, three complex deployments inspired by real data federation platforms have been conducted.

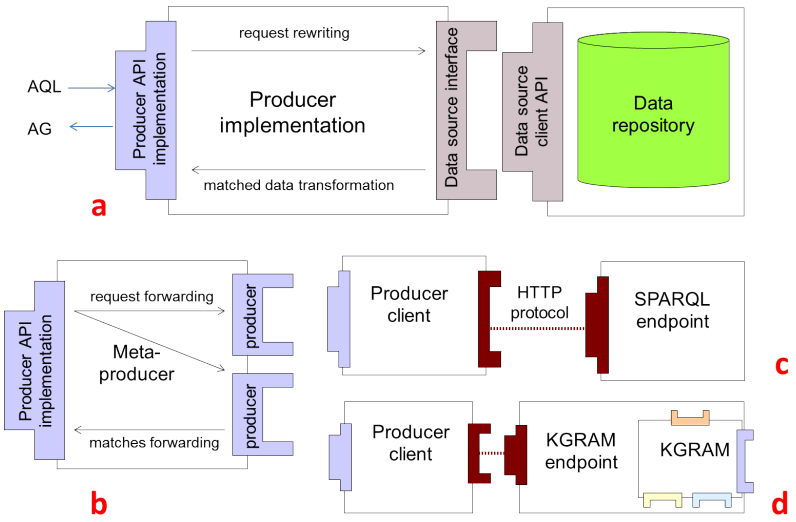

Figure 3.5: KGRAM *Producers*. Top: regular data source producer. Bottom left: meta-producer. Bottom right: producer towards remote data sources.

ISICIL. Figure 3.6 illustrates an implementation that was experimented in the context of the ISICIL project[5]. In this scenario, RDF data is distributed over three servers, each of them in charge of inferences on a specific type of data: (1) social network and user profiles, online communities, activity tracking and trust model; (2) tag model, document metadata, terminologies, thesaurus; (3) Web resource model with low level data such as MIME type, production context, format, duration, etc. A KGRAM core component is instantiated with a SPARQL *Parser*, a *Matcher* exploiting RDFS entailments, and a *Filter* supporting XSD datatypes. A Meta-Producer component is connected to this engine to interface to the three data sources, each of them interfaced through an instance of RDF *Producer*.

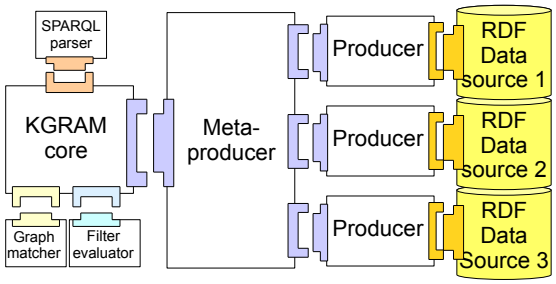

Figure 3.6: ISICIL data query architecture federating 3 RDF stores.

[5]ISICIL: `http://isicil.inria.fr`

NeuroLOG. Another complex deployment, inspired by the NeuroLOG collaborative platform for neurosciences[6], is illustrated in Figure 3.7. The use case is the joint querying of two heterogeneous data sources (an SQL and an RDF repository) located at different places, using the SPARQL query language. A SPARQL-enabled KGRAM query engine interfaced to a meta-producer is deployed. To access remote data sources, KGRAM endpoints and their associated *Producer* clients are used. Each endpoint is connected to a specific *Producer* (SQL or RDF data producer) adapting to the site data source.

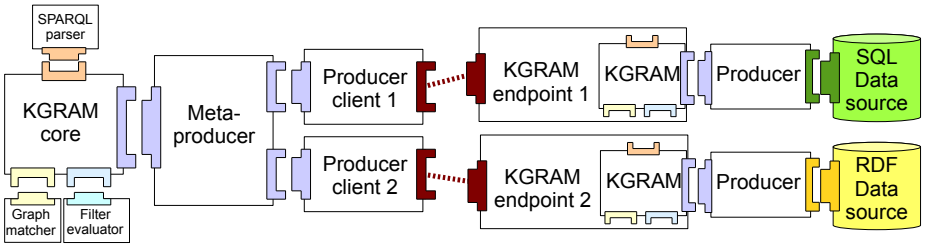

Figure 3.7: NeuroLOG data sharing architecture deployment example.

Linking Proprietary Data to Data from the LOD. In a third experiment, neuroimaging data from the NeuroLOG federation have been linked with neuroscience open knowledge capitalized through the NeuroLex initiative, allowing neuroscientists involved in NeuroLOG to benefit from the NeuroLex lexicon describing 18,490 neuroscience concepts. Thanks to KGRAM versatility, the NeuroLOG platform was easily extended with a new data source exposing the NeuroLex ontology.

3.3 Graph Rules for the Semantic Web

Rule-based modelling approaches are widely spread in many application domains and have been studied since the early age of Artificial Intelligence, for building expert systems. In the semantic Web cake, rules arrived lately, in the upper layers of the cake: the RIF rule interchange format has been recommended in 2010, 11 years after RDF, and up to now it is not really adopted. Before it, the SWRL rule language which combines OWL and a sublanguage of RuleML has been implemented and used in several projects; it is a W3C member submission dating back to 2004 but it never went further in the recommendation process. However, rules are present in many semantic Web systems — using various formats —, and they are a topic of interest since a long time in the semantic Web community. Most noticeably, Tim Berners Lee delivered a keynote speech on a Web of rules at ISWC 2005 in a joint session with the International Web Rule Symposium (RuleML)[7]. Also, the second edition of OWL recommendation in 2012 defines the OWL 2 RL profile that can be implemented

[6] NeuroLOG: `http://neurolog.polytech.unice.fr`
[7] `http://iswc2005.semanticweb.org`

with rule-based technologies and SPIN is a W3C member submission in 2011 for a SPARQL-based rule and constraint language.

KGRAM comes with a rule engine in forward chaining and another one in backward chaining and so did Corese before it. I early addressed the questions of *How should we represent rules with semantic Web standards?* and *How should we reason on RDF data with rules?* As in the rest of my work on KRR, I adopted a graph-based approach for KRR with rules.

Section 3.3.1 presents my early work on graph rules in RDF based in the CG model; Section 3.3.2 presents my work with SPARQL as rule language; Section 3.3.3 presents my contribution to the implementation of the RIF standard based on its translation into SPARQL.

3.3.1 Graph Rules based on the Conceptual Graphs Model

Graph Rules in the CG Model

Graph rules are an extension of the core CG model first introduced in [82]. A rule $R : G_1 \leftarrow G_2$ is a couple of lambda-abstractions $((\lambda x_1, \ldots x_n G_1) \leftarrow (\lambda x_1, \ldots x_n G_2))$, where G_1 and G_2 are two conceptual graphs called hypothesis and conclusion, and $x_1, \ldots x_n$ are 'connection points corresponding to n co-reference links between concepts of $G1$ and $G2$. For instance the following graph rule represents the symmetry of the relation of type `colleague`:

```
 G1 : [Person:x] -> (colleague) -> [Person:y]
=>
 G2 : [Person:y] -> (colleague) -> [Person:x]
```

Extension of RDFS with RDF Rules

In 2001, I proposed with Alexandre Delteil an extension of RDFS with rules based on the notion of existentially quantified context described in Section 3.1.2, and descending from graph rules in the CG model. In this proposal described in [51], a *graph* rule is represented in RDF, by a couple of contexts, which are two lambda abstractions representing the hypothesis and the conclusion. The RDFS vocabulary to represent rules comprises the class `Context`, the class `Axiom` subclass of `Context`, the properties `if` and `then` to link an axiom to the contexts defining its hypothesis and conclusion, the property `formalParameter` to link a variable representing a variable of an axiom shared by its hypothesis and conclusion, to the resource of type `Axiom` representing the axiom. For instance, Figure 3.8 describes a rule expressing the symmetry of property `ns:colleague`. However, as already stated in Section 3.1.2, our general proposal for representing contextual knowledge, if conceptually satisfying, was hardly tractable, implying large graphs, with many blank nodes, expensive to process and difficult to read by humans. It has not been integrated in Corese.

Corese Rule Engine for RDF

Relatedly, during the same period, I co-supervised Laurent Berthelot's master internship on the specification and implementation of the first version of the Corese rule engine to process RDF data in forward chaining. Internally, the

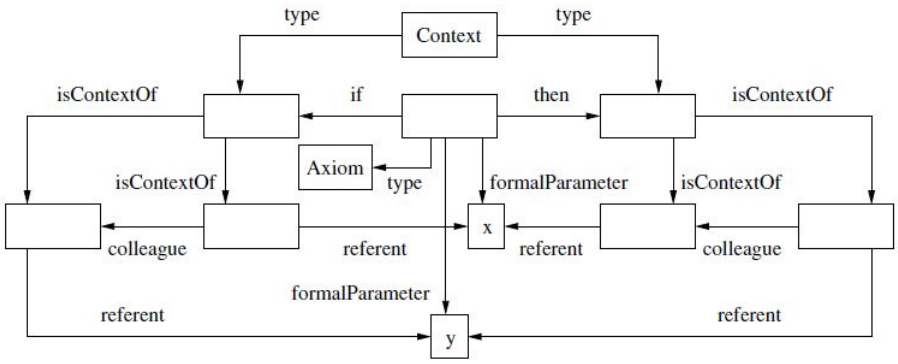

Figure 3.8: Symetry of property `ns:colleague`

engine worked on rules as defined in the CG model. We defined a simple RDF/XML rule format for Corese with a class `Rule` and two properties `if` and `then`, and syntactic conventions, e.g. to recognize rule variables [27]. This was an XML syntax rather than an RDFS schema, with implicit conventions to use it properly. The key idea, inspired by the representation of rules in CG model, was that the hypothesis and conclusion of a rule were RDF graphs.

Later on, in 2008, I co-supervised with Olivier Corby the internship of Birahim Sall who developed a forward chaining rule engine for Corese.

3.3.2 Graph Rules based on SPARQL

After the reconception of Corese into KGRAM, I co-supervised with Olivier Corby the research engineer Amel Hannech in 2011, who started the development of a new version of both rule engines, according to KGRAM architecture and API. The SPARQL rule engines in the current KGRAM distribution has been developed by Olivier Corby. From this time, we considered the SPARQL language with its CONSTRUCT query form as a rule language. The hypothesis of the rule is the WHERE clause of the SPARQL query and its conclusion is the CONSTRUCT clause. For instance, here is the representation in SPARQL of the rule stating that if a person is the brother of the parent of another person, then he is his uncle:

```
CONSTRUCT { ?x ex:uncleOf ?z }
WHERE { ?x ex:brotherOf ?y . ?y ex:parentOf ?z}
```

The semantics of the CONSTRUCT query form corresponds to forward chaining inference. To implement it, KGRAM internally manipulates the same abstract graphs for the hypothesis and conclusion of rules as those for SPARQL query graphs and RDF graphs.

In 2013, I co-supervised whith Olivier Corby the internship of Abdoul Macina who developed a compiler of SPARQL into SPIN. SPIN stands for SPARQL Inference Notation; it is a W3C member submission to represent SPARQL rules in RDF, to facilitate storage and maintenance. For instance Listing 3.2 shows the representation of the above rule in SPIN (in the Turtle syntax of RDF):

```
[ a sp:Construct ;
    sp:templates (
      [ sp:subject spin:_x ;
        sp:predicate ex:uncleOf ;
        sp:object sp:_z ] ) ;
    sp:where (
      [ sp:subject spin:_x ;
        sp:predicate ex:brotherOf ;
        sp:object sp:_y ]
      [ sp:subject sp:_y ;
        sp:predicate ex:parentOf ;
        sp:object sp:_z ] )
]
```

Listing 3.2: Representation of rule *if a person is the brother of the parent of another person, then he is his uncle* in SPIN

SPIN can be viewed as an additional syntax of the same rule language as well as a way to reify rules into RDF to process them as a special kind of knowledge (as discussed in Chapter 1). The SPARQL parser of KGRAM produces an abstract query graph which is represented as an Abstract Syntax Tree (AST). We used this AST as a pivot representation for translation of SPARQL into SPIN: Abdoul Macina specified and implemented the translation of a SPARQL AST into RDF, according to the SPIN model. The development of the inverse translation function, SPARQL2SPIN, relies on a STTL transformation presented in Section 4.2.

3.3.3 Implementation of RIF based on SPARQL

In 2009 I started working on RIF[8], the upcoming W3C recommendation for exchanging rules among Web rule engines. RIF is defined as an extensible set of dialects, three of which are defined in the recommendation: RIF-Core corresponds to the Horn Logic without function symbol, i.e to Datalog, with classical first-order logic semantics. RIF-BLD stands for Basic Logic Dialect; it corresponds to Horn Logic with equality. Syntactically, it is extended with frames, URI denoting concepts and XSD datatypes. RIF-PRD stands for Production Rules Dialect; it enables to represent production rules.

I co-supervised with Olivier Corby the internship of Corentin Follenfant in 2009 who developed a RIF parser producing an abstract syntax tree of RIF. From 2011, I co-supervised with Olivier Corby the PhD thesis of Oumy Seye [83] on the management of rules on the Web. In this framework, we worked on the identification of the RIF dialect that can be translated into SPARQL and conversely the subset of SPARQL that can be translated into RIF. Our motivation was that there is still very few implementations of RIF, whereas there is a wide range of implementations of SPARQL, among which a number of them handle SPARQL rules.

Our proposal is published in [84] and summarized in the following. RIF-BLD atomic formulas comprise positional terms $t(t_1, \ldots t_n)$ and terms with named arguments $t(s_1 -> v_1, \ldots s_n -> v_n)$ where t represents a predicate or a function, equalities, class memberships, class specializations, frames $t[p_1 -> v_1, \ldots p_n -> v_n]$,

[8] http://www.w3.org/TR/2010/NOTE-rif-overview-20100622

Table 3.3: RIF-SPARQL dialect

RIF simple terms, lists, and atomic formula		SPARQL
constant	c	c'
variable	?x	?x
closed list	$\text{List}(t_1 \dots t_m)$	$(t'_1 \dots t'_m)$
equality	$t_1 = t_2$	$\text{Filter}(t'_1 = t'_2)$
class membership	$t_1 \# t_2$	t'_1 `rdf:type` t'_2
subclass	$t_1 \#\# t_2$	t'_1 `rdfs:subclassOf` t'_2
frame	$t[p_1 -> v_1, \dots p_n -> v_n]$	$t'\ p'_1\ v'_1 \dots t'\ p'_n\ v'_n$
external term	$\text{External}(t(t_1, \dots t_n))$	$\text{Filter}(t'(t'_1, \dots t'_n))$
RIF non atomic formula		SPARQL
conjunctive formula	$And(\varphi_1, \dots \varphi_n)$	$\varphi'_1 \dots \varphi'_n$
disjunctive formula	$Or(\varphi_1, \dots \varphi_n)$	φ'_1 UNION $\dots$ φ'_n
existential formula	$Exist\ ?v_1, \dots ?v_n\ (\varphi)$	φ'
RIF-BLD rule	$Forall\ ?v_1, \dots ?v_n\ (\varphi :- \psi)$	CONSTRUCT φ' WHERE ψ'

and external terms External(φ) where φ is an atomic formula. RIF-BLD formulas comprise atomic formulas, conjunctions and disjunctions of formulas, existential formulas, rule implications, universal rules and universal facts. For instance, here is an example of a RIF-BLD universal rule with frames and conjunctions of frames:

```
Forall ?x ?y ?z (
   ?x[ex:uncleOf -> ?z] :-
   AND ( ?x[ex:brotherOf -> ?y]  ?y[ex:parentOf -> ?z] ) )
```

We showed that SPARQL queries of the CONSTRUCT form (with no OPTIONAL part in the query graph pattern) can be translated in RIF-BLD, each triple being converted into a frame, except for those with a `rdf:type` or `rdfs:subclassOf` property which are translated into class memberships and class specialization respectively, and those describing a list which are translated into a closed list. Conversely, we delineated the RIF dialect which can be translated in SPARQL as the set of RIF-BLD terms and formulas minus open lists, terms with named arguments, positional terms and universal facts, and with terms limited to constants, variables and closed lists in Equal, Member, Subclass and Frame formulas. Table 3.3 shows the terms and formula of the RIF-SPARQL dialect, that can be implemented into SPARQL. The translation of the RIF AST into the SPARQL AST of KGRAM has been developed and this made of KGRAM engine an implementation of this subset of RIF-BLD.

Conclusion

This chapter showed how my research work on knowledge representation and reasoning on the semantic Web went from considering Conceptual Graphs for the semantic Web to modelling knowledge with graphs and graph rules, while implementing W3C standards for the semantic Web. Throughout this chapter

I focused on the basic reasoning task of matching knowledge graphs, to answer (SPARQL) graph queries against (RDF) knowledge graph bases, while taking into account ontological knowledge (RDFS vocabularies or rule bases). The next chapter comes as a continuation of this one and deals with more high level reasoning tasks to answer research questions raised by the management of the linked data.

Chapter 4

Advanced Linked Data Processing

Introduction

Nowadays, the Web of data has become a reality through the publication and interlinking of various open data sets in RDF, and with the support of various initiatives such as the W3C Data Activity[1] and the Linking Open Data (LOD) project[2] aiming at Web-scale data integration and processing. Its success largely depends on the ability to reach data from "the deep Web", which keeps on growing as data is continuously being accumulated in ever more heterogeneous databases. In particular, NoSQL systems have gained a remarkable success during recent years and should be considered as potential big contributors of the linked open data. This raises the question of *How to transform relational or non relational data into RDF data?*

Conversely, in order to present RDF data to the user or feed Web applications not necessarily based on Semantic Web standards, the question which arises is *How to transform RDF data into other languages?* This is of prime interest to *present* data selected and extracted from the Web of data in a format suitable for the user (e.g., HTML or CSV). The structured Web has been provided with the XSLT transformation language to present XML data to the user into HTML pages or to transform XML data from one XML schema into another one or from an XML schema into any non XML specific text format, for XML data interchange and therefore interoperability. The Web of data now requires a transformation language for RDF, to transform RDF data from one RDF schema into another, which is already possible with SPARQL, but also to present RDF data to users or transform data from its RDF "syntax" into any other one. In fact, RDF can be viewed as a meta-model to represent on the Web other languages and models. The above research question then becomes *How to generate the concrete syntax of expressions of a given language from their RDF representation?*

Relatedly, the exploitation of the mass of RDF data now available on the semantic Web requires workable knowledge management solutions adapted to

[1] `http://www.w3.org/2013/data`
[2] `http://linkeddata.org`

the RDF model. In particular the validation of RDF data against various kinds of conditions is of prime interest to guaranty its subsequent exploitation. Just like the structured Web has been provided with the XSL Schema language to validate XML data against document models, the Web of data now requires a language to validate RDF data against structural constraint while possibly taking into account entailment regimes. The W3C RDF Data Shapes Working Group has published in 2016 a Working Draft describing the Shapes Constraint Language (SHACL), intended to become a W3C recommendation. This raises the question of *How to express constraints on RDF data?* and *How to automate the validation of RDF data against constraints?*, and in particular *How to implement the SHACL language?*

Finally, given the giant mass of RDF data now available on the semantic Web, the general question which arises is *How to automatically learn higher level knowledge from RDF data?* and in particular *How to learn ontologies from the giant mass of RDF triples available on the semantic Web?* While some epistemic communities start by producing a fine-grained ontological modelling before producing RDF data based on it, others clearly do not have such a policy or possibility, especially when their RDF data is generated from data in other formats. In the best case, the ontology structure is established, i.e. concepts are declared and hierarchically organized, and the definition of concepts remains to be done. In this case, mining RDF data is a way to automatically learn ontologies from it, or at least support the construction of ontologies.

This chapter is organized as follows: Section 4.1 summarizes my contribution to the generation of RDF data; Section 4.2 summarizes my contribution to the presentation and transformation of RDF data; Section 4.3 summarizes my contribution to the validation of RDF data; finally, Section 4.4 summarizes my contribution to ontology learning from RDF data.

The works synthesized in this chapter have been published in the proceedings of a national French conference: Journées francophones d'*Ingénierie des Connaissances* (IC 2015) [33], in a French journal: *Revue d'Intelligence Artificielle* [34], and in the proceedings of several international conferences and workshops: *Int. Conf. on Conceptual Structures* (ICCS 2002) [50], *IJCAI 2001 Workshop on Ontologies and Information Sharing* [49], *Int. Conf. on Knowledge Engineering and Knowledge Managemen*t (EKAW 2014) [91], *Int. Conf. on Knowledge Capture* (K-CAP 2015) [90], *Int. Conf. on Web Information Systems and Technologies* (WEBIST 2015, WEBIST 2016) [32][76][77], *ESWC 2015 Int. Workshop Semantic Web for Scientific Heritage* [20], *Int. Semantic Web Conferenc*e (ISWC 2015) [35], *Int. Conf. on Database and Expert Systems Applications* (DEXA 2016) [78], *Int. Conf. on Web Reasoning and Rule Systems* (RR 2016) [36].

4.1 Generating RDF Data from Heterogeneous Data

In 2014, with my colleagues Franck Michel and Johan Montagnat, we started addressing the question of transforming data from heterogeneous formats into RDF. This is the subject of Franck Michel's PhD thesis [75] which I co-supervised with Johan Montagnat. To answer this question, we proposed the xR2RML

language, an extension of R2RML, the W3C recommandation for expressing mappings from relational databases to RDF datasets. The xR2RML language may be operationalized to produce an RDF graph resulting from the transformation of the original data source, following the graph materialization approach. However, the size of some large data sets and the need for up-to-date data may require adopting the virtual graph approach. Answering this kind of use cases calls for the development of SPARQL interfaces for legacy data sources. To enable that, we proposed a two-step approach to execute SPARQL queries over heterogeneous databases. The first step consists in the definition of a pivot abstract query language and the database-independent translation of a SPARQL query in this language, based on the xR2RML mapping of the database to RDF. The second step consists in the translation of an abstract query into a concrete query by taking into account the specific query capabilities of the targeted database. We demonstrated the effectiveness of our approach by querying a MongoDB database with SPARQL queries.

Section 4.1.1 presents the xR2RML mapping language enabling both the materialization of heterogeneous data formats into RDF or the rewriting of SPARQL queries on a virtual RDF graph to dynamically access legacy data. Section 4.1.2 presents our approach to translate SPARQL queries into heterogeneous query languages via an abstract query language. These sections summarize our contributions published in [76], [77] and [78].

4.1.1 The xR2RML Mapping Language

An R2RML mapping is an RDF graph representing *triples maps*, each one specifying how to map rows of a logical table from a relational database to RDF triples.

Basically, xR2RML triples maps extend R2RML triples maps by referencing a *logical source* which is the result of a request applied to the input database. It is either a *base table* or a *view*. The xR2RML base table extends the concept of R2RML *table or view* beyond relational databases, to tabular databases (extensible column store, CSV/TSV, etc.). An xR2RML view represents the result of executing a query against the input database. It is associated to a query expression with a property `query` that extends `rr:sqlQuery`. This must be a valid expression with regards to the query language supported by the input database but no other assumption is made to the query language. Retrieving values from a query result set requires evaluating data element references against the query result. xR2RML uses the concept of *reference formulation* of a logical source to name the syntax of data element references. An xR2RML processor must be provided with a database connection and the reference formulation applicable to results of queries run against the connection. In addition, xR2RML enables to generate RDF collections or containers from one-to-many relations modelled as compound values or as cross-references and to perform joint queries following cross-references between logical resources.

For instance, Listing 4.1 shows an example MongoDB database with one collection containing two JSON documents describing the two projects held in a company. Each project is described by a name, a code and a set of teams. Each team is an array of members, and we assume that the last member is always the team leader.

```
{ "project":"Finance & Billing", "code":"fin",
  "teams":[ [ {"name":"P. Russo"}, {"name":"F. Underwood"} ],
            [ {"name":"R. Danton"}, {"name":"E. Meetchum"} ] ] },
{ "project":"Customer Relation", "code":"crm",
  "teams":[ [ {"name":"R. Posner"}, {"name":"H. Dunbar"} ] ] }
```

Listing 4.1: An example MongoDB database with one collection containing two JSON documents describing the two projects held in a company. Each project is described by a name, a code and a set of teams. Each team is an array of members

Listing 4.2 shows an xR2RML mapping graph describing a mapping identified by `<#TmLeader>`. The logical source is the MongoDB query `"db.projects.find({})"` that simply retrieves all documents from collection `"projects"`. The mapping associates team leaders (object) to projects (subject) with predicate `ex:teamLeader`. This is done by means of a JSONPath expression that selects the last member of each team using the calculated array index `"[(@.length - 1)]"`.

```
<#TmLeader>
  xrr:logicalSource [xrr:query "db.projects.find({})"];
  rr:subjectMap [rr:template
    "http://example.org/project/{$.code}".];
  rr:predicateObjectMap [
    rr:predicate ex:teamLeader;
    rr:objectMap [ xrr:reference
      "$.teams[0,1][(@.length − 1)].name" ] ].
```

Listing 4.2: An example xR2RML mapping graph describing a mapping

xR2RML mapping language has been validated in a prototype implementation supporting several RDBs and the MongoDB NoSQL document store. It has already been used to transform the JSON export of the TAXREF[3] taxonomy into a (materialized) SKOS vocabulary [20].

4.1.2 xR2RML-based SPARQL Query Rewriting

Various methods have been defined to translate SPARQL queries into another query language, that are generally tailored to the expressivity of the target query language. Notably, the rich expressivity of SQL and XQuery makes it possible to define semantics-preserving SPARQL rewriting methods. By contrast, NoSQL databases typically trade expressivity for scalability and fast retrieval of denormalised data. For instance, many of them hardly support joins. Therefore, to envisage the translation of SPARQL queries in the general case, we propose a two-step method. A SPARQL query is first rewritten into a pivot abstract query under xR2RML mappings, independently of any target database, then the pivot query is translated into concrete database queries based on the specific target database capabilities and constraints.

Translating SPARQL Queries into Abstract Queries under xR2RML Mappings

Listing 4.3 shows the grammar of our pivot abstract query language:

[3] `https://inpn.mnhn.fr/programme/referentiel-taxonomique-taxref`

```
<AbsQuery> ::= <Query> | <Query> FILTER <filter> | <AtomicQuery>
<Query> ::=
  <AbsQuery> INNER JOIN <AbsQuery> ON {v1,...vn} |
  <AbsQuery> AS child INNER JOIN <AbsQuery> AS parent
             ON child/<Ref> = parent/<Ref> |
  <AbsQuery> LEFT OUTER JOIN <AbsQuery> ON {v1,...vn}|
  <AbsQuery> UNION <AbsQuery>
<AtomicQuery> ::= {From, Project, Where}
```

Listing 4.3: Grammar of the pivot abstract query language (AQL)

Operators INNER JOIN ON, LEFT OUTER JOIN ON and UNION are entailed by the dependencies between graph patterns of the SPARQL query, and SPARQL filters involving variables shared by several triple patterns result in a FILTER operator. The computation of these operators is delegated to the target database if it supports them (i.e. if the target query language has equivalent operators like in the case of a relational database), or to the query processing engine otherwise (e.g. MongoDB cannot process joins). Each SPARQL triple pattern *tp* is translated into a union of atomic abstract queries (`<AtomicQuery>`), under the set of xR2RML mappings likely to generate triples matching *tp*. The components of an atomic abstract query are as follows:

- `From` is the xR2RML mapping's logical source (the value of property `xrr:query`) and its optional iterator (the value of property `rml:iterator`).

- `Project` is the set of xR2RML references that must be projected, i.e. returned as part of the query results. In the relational case, projecting an xR2RML reference simply means that the column name shall appear in the SQL SELECT clause. When dealing with MongoDB, it amounts to projecting the JSON fields mentioned in the JSONPath reference.

- `Where` is a conjunction of abstract conditions entailed by matching each term of triple pattern *tp* with its corresponding term map in an xR2RML mapping *TM*. Three types of condition may be created:
(i) a SPARQL variable in the triple pattern is turned into a not-null condition on the xR2RML reference corresponding to that variable in the term map, denoted by `isNotNull(<`*xR2RML reference*`>)`;
(ii) A constant triple pattern term (IRI or literal) is turned into an equality condition on the xR2RML reference corresponding to that RDF term in the term map, denoted by `equals(<`*xR2RML reference*`>,`*value*`)`;
(iii) A SPARQL filter condition f on a SPARQL variable is turned into a filter condition, denoted by `equals(<`*xR2RML reference*`>,`f`)`.
For instance, Listing 4.4 shows an example SPARQL query and its translation in the abstract query language:

```
SELECT ?proj WHERE {?proj ex:teamLeader "H. Dunbar".}

{ From: {"db.projects.find({})"},
  Project: {$.code AS ?proj},
  Where:
  { isNotNull($.code),
    equals($.teams[0,1][(@.length−1)].name, "H. Dunbar") }}
```

Listing 4.4: An example SPARQL query and its translation in the abstract query language

The `From` part of the abstrat query is the query in the logical source of the xR2RML mapping `<#TmLeader>`. In its `Project` part, variable `?proj` is associated to the subject map's reference of the xR2RML mapping since it is the subject of the SPARQL triple pattern. The conditions in the `Where` part are calculated by matching each term of the SPARQL triple pattern with its corresponding term map in the xR2RML mapping `<#TmLeader>`.
The JSON documents needed to answer this abstract query should verify the two conditions in its `Where` part. In the next section, we present our method to rewrite such conditions into concrete MongoDB queries.

Translating an Abstract Query into a Concrete Query Language

The second step of our approach to dynamically access various types of data format with SPARQL consists in translating the abstract pivot query into concrete queries by taking into account the specific query capabilities of the targeted database. We demonstrated the effectiveness of our approach by querying a MongoDB database with SPARQL. MongoDB provides a JSON-based declarative query language consisting of two major mechanisms: the *find* query retrieves documents matching a set of conditions; the *aggregate* query allows for the definition of processing pipelines. In our work we considered the *MongoDB query language* limited to *find* queries. We defined a set of rules to translate the pivot abstract query into an abstract representation of a MongoDB query, and we showed that the latter can always be rewritten into a union of concrete MongoDB *find* queries that shall return all the documents required to answer the SPARQL query. Both the abstract representation of a MongoDB query, the translation rules of the abstract query language into this abstract representation of the MongoDB query language and the final translation into the concrete MongoDB query language are presented in [78] and [75] and an open source prototype implementation has been developed by Franck Michel[4]

Due to the limited expressivity of the MongoDB *find* queries, some JSONPath expressions cannot be translated into equivalent MongoDB queries. Consequently, the query translation method cannot guarantee that query semantics be preserved. Yet, we ensure that rewritten queries retrieve all matching documents, possibly with additional non-matching ones. The RDF triples thus extracted are subsequently filtered by evaluating the original SPARQL query. This preserves semantics at the cost of an extra SPARQL query evaluation.

4.2 Transforming RDF data into presentation formats or other data formats

In 2014, with my colleague Olivier Corby, we started addressing the question of transforming RDF data into other data formats. With the growing number and variety of RDF datasets now available on the Web comes the need to transform RDF data, to present RDF data to the user or to provide input data to Web services consuming other data formats, e.g. JSON, XML, CSV, etc. To answer this question, we specified and implemented the SPARQL Template Transformation Language (STTL) and we developed several STTL-based transformations

[4] `https://github.com/frmichel/morph-xr2rml`

to answer the various above cited scenarios. Section 4.2.1 presents the STTL language and Section 4.2.2 presents two illustrative STTL-based transformers. These sections summerize our contributions published in [32], [33], [34] and [35].

4.2.1 SPARQL Template Transformation Language

STTL is a generic and domain independent extension to SPARQL supporting the declarative representation of any special-purpose RDF transformation as a set of transformation rules. We conceived it as a lightweight syntactic extension to SPARQL that can be compiled into standard SPARQL. As a result, STTL inherits SPARQL's expressivity and extension mechanisms.

More precisely, STTL relies on two extensions of SPARQL: an additional TEMPLATE query form to express transformation rules and extension functions to recursively call the processing of a template into another one. A TEMPLATE query is made of a standard WHERE clause and a TEMPLATE clause. The WHERE clause is the condition part of a rule, specifying the nodes in the RDF graph to be selected for the transformation. The TEMPLATE clause is the presentation part of the rule, specifying the output of the transformation performed on the solution sequence of the condition. For instance, let us consider the OWL axiom stating that the class of parents is equivalent to the class of individuals having a person as child. Here are its expressions in Functional syntax:

```
EquivalentClasses(
  a:Parent
  ObjectSomeValuesFrom(a:hasChild a:Person))
```

and in Turtle:

```
a:Parent a owl:Class ;
  owl:equivalentClass
    [ a owl:Restriction ;
      owl:onProperty a:hasChild ;
      owl:someValuesFrom a:Person ]
```

Listing 4.5 shows the STTL template enabling to transform the above `equivalentClass` statement from RDF into Functional syntax.

```
TEMPLATE {
 "EquivalentClasses("
 st:apply-templates(?in) " "
 st:apply-templates(?c) ")" }
WHERE { ?in owl:equivalentClass ?c . }
```

Listing 4.5: An example STTL template enabling to transform an `equivalentClass` OWL statement from RDF into Functional syntax

The value matching variable `?in` is `a:Parent` which is expected in the transformation output (the Functional syntax of the OWL 2 statement), while the value matching variable `?c` is a blank node whose property values are used to build the expected output. This is defined in another template to be applied on this focus node. The `st:apply-templates` extension function enables this recursive call of templates, where `st` is the prefix of STTL namespace[5].

[5] `http://ns.inria.fr/sparql-template`

More generally, `st:apply-templates` function can be used in any template t_1 to execute another template t_2 that can itself execute a template t_3, etc. Hence, templates call themselves one another, in a series of call, enabling a hierarchical processing of templates and a recursive traversing of the target RDF graph. Similarly, `st:call-template` function can be used to call named templates.

STTL is compiled into standard SPARQL. This allows the approach to be usable with different implementations of the standard, to benefit from its expressivity, from the native extension mechanisms and also from the optimizations of the implementations. The compilation keeps the WHERE clause, the solution modifiers and the VALUES clause of the template unchanged and the TEMPLATE clause is compiled into a SELECT clause. For instance, the TEMPLATE clause of the following STTL template:

```
TEMPLATE {
  "ObjectSomeValuesFrom(" ?p " " ?c ")" }
WHERE {
  ?in a owl:Restriction ;
    owl:onProperty ?p ;
    owl:someValuesFrom ?c }
```

is compiled into the following standard SPARQL SELECT clause:

```
SELECT
  (CONCAT("ObjectSomeValuesFrom(",
   st:process(?p), " ",
   st:process(?c), ")") AS ?out)
```

Since STTL can be compiled into SPARQL 1.1, the evaluation semantics of a template is that of SPARQL.

An STTL engine has been implemented within the Corese semantic Web Factory[6]. It comprises an `STTL` RESTful Web service to process STTL transformations on local or distant RDF dataset. Basically, the algorithm of the STTL engine is as follows. The template processor is called by `st:apply-templates` or other alike extension functions. Given an RDF graph with a focus node to be transformed and a list of templates, it successively tries to apply them to the focus node until one of them succeeds. A template succeeds if the matching of the WHERE clause succeeds, i.e., returns a result. If no template succeeds, a default template is applied to the focus node. Recursive calls to `st:apply-templates` within templates implement the graph recursive traversal with successive focus nodes.

4.2.2 STTL-based RDF Transformers

An STTL engine is generic: it applies to any RDF data with any set of STTL templates. What is specific to each transformation is the set of STTL templates defining it. In other words, each RDF transformer specific to an output format is defined by a specific set of STTL templates processed by the generic template processor implementing STTL.

[6] `http://wimmics.inria.fr/corese`

We defined several such STTL-based RDF transformers, among which: (1) RDF-to-RDF transformers for the transformation of RDF data from any RDF syntax into any other RDF syntax, e.g., RDF/XML-to-Turtle; (2) RDF-to-HTML transformers enabling to design Linked Data navigators; (3) RDF-syntax-to-anotherSyntax transformers, e.g., a transformer of OWL 2 statements from the OWL/RDF syntax into the OWL 2 Functional Syntax, and a transformer of SPARQL queries in SPIN RDF syntax into SPARQL concrete syntax. They all are available online[7] and a demo transformation service is available online[8]. Here we give an overview of two of these transformers.

STTL-based Linked Data Navigators

The keys to build a Linked Data navigator is (1) to generate HTML pages for RDF resources and (2) to generate hyperlinks in the output HTML code. This is achieved with *href* attributes having as value a URL conveying a request for STTL transformation to the transformation service. Here is an example of a named STTL template to construct a hyperlink to a focus URI `?x`:

```
TEMPLATE st:link(?x) {
 "<a href='/template?profile=st:dbpedia&uri=" str(?x) "'>" str(?x) "</a>"}
WHERE { }
```

In the URL constructed by the above template to convey the request for a STTL transformation, the URI of a *profile* is indicated to the STTL service as the value of a `profile` key. We introduced this notion of *profile* of a transformation to associate a SPARQL query and an STTL transformation into a simple workflow; it is detailed in [35].
When applied on a given URI the above template would produce for instance the following output code:

```
<a href='/template?profile=st:dbpedia&uri=http://fr.dbpedia.org/resource/
Antibes'>http://fr.dbpedia.org/resource/Antibes</a>
```

Based on these principles, among others, we developed a domain-specific Linked Data navigator — a server with its STTL service and the `st:navlab` RDF-to-HTML transformation — to browse the DBpedia dataset, specifically on persons and places. Figure 4.1 is the screenshot of an HTML page produced by this navigator. We wrote the `st:navlab` transformation as a set of 24 STTL templates which are available online[9]. Here is a template in `st:navlab`, to construct the table of resource descriptions; it recursively calls the `st:title` named template to output the title in HTML and the `st:descresource` to build the description of each resource selected in DBpedia.

```
TEMPLATE {
  st:call-template(st:title, ?in, ?label, (coalesce(?ic, "")))
  "<table>" st:call-template(st:descresource, ?in) "</table>" }
WHERE {
  ?in a <http://dbpedia.org/ontology/Resource> .
  ?in rdfs:label ?label FILTER(lang(?label) = 'fr')
  OPTIONAL { ?in <http://dbpedia.org/ontology/thumbnail>  ?ic } }
```

[7] `https://ns.inria.fr/sparql-template/`
[8] `http://corese.inria.fr`
[9] `http://ns.inria.fr/sparql-template/navlab`

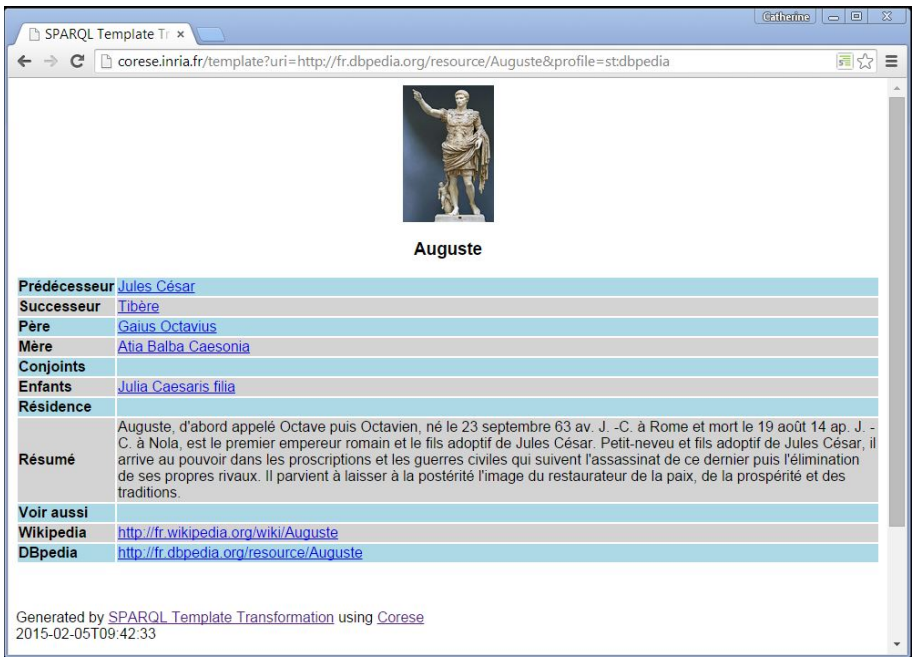

Figure 4.1: DBpedia Navigator

The DBpedia SPARQL endpoint is accessed through a SERVICE clause in a predefined CONSTRUCT query to retrieve relevant data according to the types of resources that the application is interested in: our navigator focuses on historical people and places. Then the `st:navlab` transformation is applied to the resulting RDF graph. It generates a presentation in HTML format of the retrieved data, adapted to the type of targeted resources — people and places. In particular, the transformation localizes places on a map.

As it can be viewed in Figure 4.1, when following the hyperlink generated by the DBpedia navigator, a request is sent to the STTL server to produce an HTML presentation of the DBpedia resource on Augustus, according to the `st:dbpedia` profile (embedding the `st:navlab` transformation).

The interest of this STTL-based approach of DBpedia-to-HTML transformation is that it is declarative and can therefore easily be extended to handle the presentation of other types or resources by adding new dedicated templates.

STTL-based RDF-to-Any Other Syntax Transformers

Among others, we developed an STTL-based transformer of OWL 2 statements from the OWL/RDF syntax into the OWL 2 Functional Syntax. The transformation follows the W3C Recommentation *OWL 2 Web Ontology Language Mapping to RDF Graphs*[10]. We wrote it as a set of 73 STTL templates, structured into 5 subsets, available online[11]. We validated the `st:owl` transformation on the OWL 2 Primer ontology[12] containing 350 RDF triples. Let us note that

[10] `http://www.w3.org/TR/owl2-mapping-to-rdf`
[11] `http://ns.inria.fr/sparql-template/owl`
[12] `http://www.w3.org/TR/owl2-primer`

the results are equivalent but not identical because some statements are not printed in the same order, due to the fact that Protégé does not save RDF/XML statements exactly in the same order and hence blank nodes are not allocated in the same order. and we tested this transformer on several real world ontologies, among which a subset of the *Galen* ontology. The RDF graph representing it contains 33,080 triples, the size of the result is 0.58 MB and the (average) transformation time is 1.75 seconds. We also tested it on the *HAO* ontology. The RDF graph representing it contains 38,842 triples, the size of the result is 1.63 MB, the (average) transformation time is 3.1 seconds.

In the continuation of our work on STTL, we considered answering other Linked Data management issues with this language, among which the validation of RDF data against constraints. This is discussed in the next section. In addition, we considered generalizing the STTL language to provide the semantic Web with a script language. This is discussed in conclusion.

4.3 Validating RDF Data against Constraints

In 2015, with my colleague Olivier Corby, we started addressing the question of expressing constraints on RDF data and checking that an RDF graph satisfies some given constraints. This is a key issue for the development of full-fledged Linked Data based solutions. To answer this question we proposed an approach based on the STTL language. We showed that STTL, originally designed to transform RDF data into any data format, can also be used as a constraint language for RDF: each STTL template is viewed as representing a constraint and an RDF graph is checked against a set of constraints by applying the set of STTL templates representing these constraints on the RDF graph. The output of the application of a set of STTL templates can be a simple boolean value or a convenient textual view of the data, where for instance, the subgraphs violating the constraints are highlighted.

As an interesting special use case, we applied our approach to ontology validation, which is a key issue in ontology engineering. We implemented the semantics of OWL 2 profiles as sets of constraints formalized in STTL and we checked OWL ontologies in RDF syntax against these constraints to characterize their expressivity. This contribution has been published in [36] and is summarized in the following. Section 4.3.1 presents our STTL-based approach to validate ontologies against OWL 2 profiles and Section 4.3.2 describes our approach to visualize the result of such a validation in the same integrated framework.

4.3.1 STTL-based Validation of Ontologies against OWL Profiles

As stated in the W3C recommendation, each OWL 2 profile is defined as a set of restrictions on the structure of OWL 2 statements, i.e. syntactic constraints on OWL 2 axioms definitions[13]: (1) a set of restrictions on the type of class expressions that can be used in axioms and on the place in which they can be used, (2) the set of OWL axioms supported when restricted to the allowed set of

[13] `https://www.w3.org/TR/owl2-profiles/`

class expressions, (3) the set of OWL constructs which are not supported. For example, in OWL 2 RL, the constructs in the subclass and superclass expressions in `SubClassOf` axioms must follow some usage patterns and OWL 2 RL axioms are undirectly constrained by these restrictions.

We defined an STTL transformation to represent each of the three OWL 2 profiles defined in the W3C recommendation. Each STTL template participating to these transformations enables to check a specific OWL 2 model constraint and returns a boolean value. When traversing the RDF graph representing the ontology to be validated against a given OWL 2 profile, the boolean results of the templates applied to the graph nodes are aggregated by using a logical conjunction instead of a string concatenation, so that the final result is a boolean value indicating whether the profile checking succeeds or fails.

For instance let us consider the `st:owlrl`[14] transformation which comprises 36 STTL templates representing the constraints defining the OWL 2 RL profile. It consists of a start template calling the `st:axiom` transformation whose templates themselves call the `st:subexp`, `st:superexp`, and `st:equivexp` transformations. Transformation `st:axiom` comprises 10 templates representing restrictions on class axioms to use the appropriate form of class expressions, restrictions on property domain and range axioms to only use class expressions of type `superClassExpression`, restriction on positive assertions to only use class expressions of type `superClassExpression` and restrictions on keys to only use `subClassExpression`. The following example template represents the restriction on `subClassOf` axioms to use a class expression of type `superClassExpression` (respectively `subClassExpression`) for the superclass (respectively the subclass). These two types of class expressions are each defined by another STTL transformation which is recursively called in the WHERE clause of the template. Both transformations return a boolean whose value corresponds to the conformance of the class expressions. The template returns the conjunction of these two values.

```
TEMPLATE { ?suc }
WHERE {
  ?in rdfs:subClassOf ?y
  BIND (
   st:call-template-with(st:subexp, st:subClassExpression, ?in) &&
   st:call-template-with(st:superexp, st:superClassExpression, ?y)
  AS ?suc)
  FILTER st:alreadyVisited(?in, "subClass", ?suc) }
```

In addition, `st:axiom` comprises one template representing the disallowance of the `DisjointUnion` axiom and of reflexive properties. This template returns `false` if such an axiom or property occurs in the ontology:

```
TEMPLATE { false }
WHERE {
  {?in owl:disjointUnionOf ?y} UNION {?in a owl:ReflexiveProperty}
  FILTER (st:alreadyVisited(?in, "fail", false)) } LIMIT 1
```

[14] `http://ns.inria.fr/sparql-template/owlrl/owlrl`

4.3.2 STTL-based Vizualisation of Validation Results

In order to provide the user with a visualization of the result of the validation, we wrote an STTL transformation to present in a HTML document the RDF graph (in the Turtle syntax) representing the ontology to be validated, where non valid triples are highlighted. For instance, figure 4.2 shows the visualization of an ontology represented in Turtle and tested against the OWL 2 RL profile with `owl:complementOf` in red since OWL 2 RL does not allow it within a class intersection inside a class equivalence.

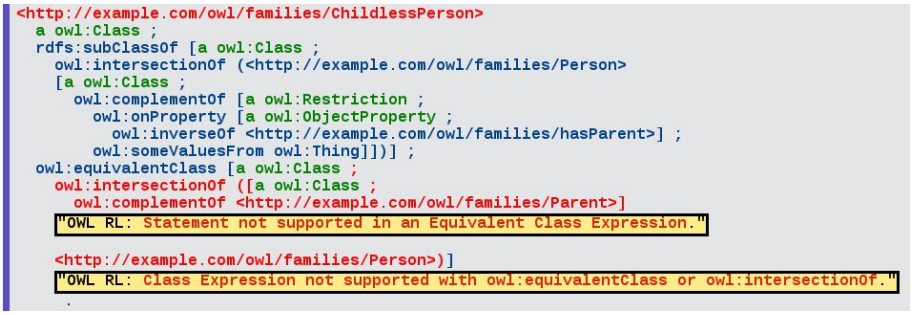

Figure 4.2: Visualizing the validation result of an ontology against OWL 2 RL

During the traversal of the RDF graph representing the tested ontology, a visitor records the subjects of RDF triples corresponding to failing statements. After type check resumes, the visitor is given to an STTL transformation RDF2Turtle which enables to pretty-print RDF graphs in Turtle. The template below is the key of the STTL transformation. It uses the `st:visited(?in)` extension function which returns true if the node has been visited (and hence represents a failing statement). When processing a node of the RDF graph representing the vocabulary to be validated, in case this node represents a failing OWL statement, the STTL template generates an HTML element to embed the transformation of the node, i.e. its pretty-print in Turtle embedded in HTML: `<span class='fail'>...</span>`. A CSS stylesheet associates a specific presentation format to the `fail` class, e.g. a red font color.

```
TEMPLATE { FORMAT {
  if (st:visited(?in), "[<span class='fail'>%s</span>].", "[%].")
  ibox { st:call-template(st:type, ?in)
         st:call-template(st:value, ?in) } }}
WHERE { ?in ?p ?y  FILTER isBlank(?in) } LIMIT 1
```

In the continuation of our work on ontology validation, we are currently working on the general issue of RDF constraint checking and we consider the implementation of SHACL, the upcoming W3C standard, with STTL. This is further discussed in conclusion.

4.4 Ontology Learning from the Web of Data

Nowadays, ontology learning is quite a hot research topic in the semantic Web community. Back to 2001, at the time of my first work on this topic, with

Alexandre Delteil and Rose Dieng, it already appeared among the research topics of the KRR communities but our work was one of the first contributions in the semantic Web community and our paper published in the IJCAI workshop proceedings [49], although describing a modest contribution is one of my most cited articles. At that time, we addressed the general issue of ontology learning by answering the more specific question of *How should we conceptually cluster the semantically annotated resources of an epistemic community?* Our approach is summarized in Section 4.4.1.

In 2014, I started working again on ontology learning with my colleagues Andrea Tettamanzi and Fabien Gandon. Time has passed and we now address the question of *How should we automatically learn ontology from the giant mass of RDF triples available on the LOD?* by adopting a general approach consisting in automatically generating candidate OWL axioms and testing them against the facts published on the LOD. Our contributions have been published in [91] and [90] and is summarized in Section 4.4.2.

4.4.1 Concept Formation and Conceptual Clustering of Resources

In the framework of Alexandre Delteil's PhD thesis [47], we proposed a method for learning concepts from the RDF dataset gathering the semantic annotations of resources and organize these annotated resources into a conceptual hierarchy, with the ultimate goal of improving Information Retrieval in the corporate memory. We adopted an approach of concept formation where each concept is defined in extension by a subset of resources and in intention by a description shared by these resources. In our approach, we systematically consider all the possible sets of resources sharing a common description. As a result, we build a concept *lattice*, which nodes are partially ordered by the inclusion relation on their extensions, as well as by the generalisation relation on their intentions.

Our approach of concept formation was slightly different from state-of-the-art concept formation approaches in that it aimed at *systematically* generating a class for each possible set of objects, and thus building a concept *lattice* instead of choosing classes according to a given criterion and building a particular concept hierarchy. Our systematic approach was inspired from formal concept analysis and knowledge organization.

In order to deal with the intrinsic complexity of building a generalization hierarchy, we proposed an incremental approach of conceptual clustering consisting in gradually increasing the size of the resource descriptions, i.e. the maximal length of a path in the subgraphs of the RDF dataset taken as resource descriptions [49][50]. To build a concept hierarchy L_1 based on resource descriptions of length 1, concepts are created by matching resource descriptions two by two and by generalizing them. At each step, the non maximal subsets of resources are discarded. Then, building a concept hierarchy L_n based on resource descriptions of length n relies on the concept hierarchy L_{n-1} and the concept hierarchy L_1: concept descriptions of length n are built by joining all the possible pairs of one concept description of length $n-1$ in L_{n-1} and one concept description of length 1 in L_1. We tested our approach in the framework of the CoMMA European IST project dedicated to ontology-based Information Retrieval in a corporate memory.

4.4.2 Automatic Axiom Induction from RDF Data

Twelve years later, I came back again to the question of ontology learning from RDF data. In 2014, with my colleagues Andrea Tettamanzi and Fabien Gandon, we proposed an approach for the automatic induction of OWL 2 axioms from RDF data, based on (candidate) axiom scoring, in order to provide a basis for ontology learning.

Axiom Scoring Heuristics

The axiom scoring heuristics we proposed is based on possibility theory. Starting from the founding principle of possibility theory that a hypothesis should be regarded as all the more *necessary* as it is explicitly supported by facts and not contradicted by any fact, and all the more *possible* as it is not contradicted by facts, we defined an axiom scoring heuristics combining the necessity and possibility of an axiom, which are themselves defined based on the number of confirmations and the number of counterexamples of an axiom.

More precisely, we defined the *content* of an axiom ϕ, $content(\phi)$, as the finite set of formulas constructed from the set-theoretic formulas expressing the semantics of ϕ by "grounding" them, this set being restricted to just those ψ which can be counterexamples of ϕ, thus leaving out all those ψ which would be trivial confirmations of ϕ. For instance, let us consider the following candidate OWL axiom:

$$\phi = \texttt{SubClassOf(dbo:LaunchPad dbo:Infrastructure)},$$

Its content is defined as follows:

$$content(\phi) = \{ \quad \texttt{dbo:LaunchPad}(r) \Rightarrow \texttt{dbo:Infrastructure}(r) : \texttt{dbo:LaunchPad(r) is in the dataset} \quad \}.$$

We denote by u_ϕ the cardinality of $content(\phi)$, by u_ϕ^+ the number of formulas $\psi \in content(\phi)$ which are entailed by the RDF dataset (confirmations), and by u_ϕ^- the number of such formulas whose negation $\neg\psi$ is entailed by the RDF dataset (counterexamples). Then we define the possibility Π and necessity N of an axiom ϕ as follows:

$$\Pi(\phi) = 1 - \sqrt{1 - \left(\frac{u_\phi - u_\phi^-}{u_\phi}\right)^2};$$

$$N(\phi) = \begin{cases} \sqrt{1 - \left(\frac{u_\phi - u_\phi^+}{u_\phi}\right)^2}, & \text{if } u_\phi^- = 0, \\ 0, & \text{if } u_\phi^- > 0. \end{cases}$$

Finally we combine the possibility and necessity of an axiom to define a single handy acceptance/rejection index (ARI) as follows:

$$\mathrm{ARI}(\phi) = N(\phi) - N(\neg\phi) = N(\phi) + \Pi(\phi) - 1 \in [-1, 1].$$

Listing 4.6 shows the general algorithm for the overall axiom scoring process.

compute u_ϕ and u_ϕ^+;
if $0 < u_\phi^+ \leq 100$ then query a list of confirmations;
if $u_\phi^+ < u_\phi$ then
  compute u_ϕ^-;
  if $0 < u_\phi^- \leq 100$ then query a list of counterexamples;
else $u_\phi^- \leftarrow 0$;
compute $\Pi(\phi)$, $N(\phi)$ and finally $ARI(\phi)$.

Listing 4.6: Axiom scoring algorithm

A Framework for Candidate Axiom Testing

We have developed a framework for testing candidate OWL 2 axioms against a given RDF store based on the above described scoring heuristics, which uses the model-theoretic semantics of OWL 2 and SPARQL queries. Up to now, we restricted our attention to subsumption axioms involving atomic classes. Scoring `SubClassOf` axioms with their ARI requires to compute the interpretation of `Class` and `ObjectComplementOf` class expressions.

Computational definition of `Class` and `ObjectComplementOf` class expressions. We define a mapping $Q(E, \texttt{?x})$ from OWL 2 class expressions to SPARQL graph patterns, where E is an OWL 2 class expression, and `?x` is a variable, such that the query `SELECT DISTINCT ?x WHERE {` $Q(E, \texttt{?x})$ `}` returns all the individuals which are instances of E. We denote this set by $[Q(E, \texttt{?x})]$:

$$[Q(E, \texttt{?x})] = \{v : (?x, v) \in \mathit{ResultSet}(\texttt{SELECT DISTINCT ?x WHERE}\{Q(E, \texttt{?x})\}\}.$$

For a `Class` class expression A, $Q(A, \texttt{?x}) = \{\texttt{?x a } A\}$, where A is a valid IRI.

For an `ObjectComplementOf` class expression, things are slightly more complicated, since RDF does not support negation. To learn axioms from an RDF dataset, the open-world hypothesis must be made: the absence of supporting evidence does not necessarily contradict an axiom, moreover an axiom might hold even in the face of a few counterexamples. Therefore, we define $Q(\neg C, \texttt{?x})$ as follows, to approximate an open-world semantics:

$$Q(\neg C, \texttt{?x}) = \{\texttt{?x a ?dc . FILTER NOT EXISTS}\{\texttt{?z a ?dc.}Q(C, \texttt{?z})\}\},$$

where `?z` is a variable that does not occur anywhere else in the query.
For an atomic class expression A, this becomes:

$$Q(\neg A, \texttt{?x}) = \{\texttt{?x a ?dc . FILTER NOT EXISTS}\{\texttt{?z a ?dc . ?z a } A\}\},$$

Computational definition of the support and the ARI of `SubClassOf` axioms. According to the definition of the support of an axiom and following the principle of selective confirmation,

$$u_{C \sqsubseteq D} = \|\{D(a) : C(a) \text{ in the RDF dataset}\}\|,$$

because, if $C(a)$ holds, then $C(a) \Rightarrow D(a) \equiv D(a)$.
As a result, a computational definition of $u_{C \sqsubseteq D}$ is the following SPARQL query:

$$\texttt{SELECT (count(DISTINCT ?x) AS ?u) WHERE}\{Q(C, \texttt{?x})\}.$$

In order to compute the score of `SubClassOf` axioms, $ARI(C \sqsubseteq D)$, we must provide a computational definition of $u^{+}_{C \sqsubseteq D}$ and $u^{-}_{C \sqsubseteq D}$. We start with the following statements:

- confirmations are individuals $i \in [Q(C, \texttt{?x})] \cap [Q(D, \texttt{?x})]$;
- counterexamples are individuals $i \in [Q(C, \texttt{?x})] \cap [Q(\neg D, \texttt{?x})]$.

This may be translated into the following two SPARQL queries to compute $u^{+}_{C \sqsubseteq D}$ and $u^{-}_{C \sqsubseteq D}$ respectively:

```
SELECT (count(DISTINCT ?x) AS ?nConfirm)
WHERE { Q(C, ?x) Q(D, ?x) }
```

and

```
SELECT (count(DISTINCT ?x) AS ?nCounter)
WHERE { Q(C, ?x) Q(¬D, ?x) }.
```

Notice that an i such that $i \in [Q(C, \texttt{?x})]$ and $i \notin [Q(D, \texttt{?x})]$ does not contradict $C \sqsubseteq D$, because it might well be the case that the assertion $D(i)$ is just missing. Likewise, an $i \in [Q(\neg D, \texttt{?x})]$ such that $i \in [Q(\neg C, \texttt{?x})]$ will not be treated as a confirmation, based on our choice to regard as evidence in favour of a hypothesis only *selective* confirmations.

Scalable Axiom Scoring based on Time Prediction

Our proposed scoring heuristic described above is much heavier, from a computational point of view, than state-of-the-art probabilistic scoring. Fortunately, there was an evidence in our experiments that the time it takes to test an axiom tends to be inversely proportional to its score. This suggested that time-capping the test of an axiom might be an acceptable additional heuristic to decide whether to accept or reject a candidate axiom, for an axiom which takes too long to test will likely end up having a very negative score. As a result, we defined two heuristics to scale axiom scoring: The first one is based on the empirical findings that the time it takes to test a candidate `SubClassOf` axiom of the form $C \sqsubseteq D$ tends to be proportional to the product of its support $u_{C \sqsubseteq D}$ or sample size, i.e., the cardinality of the set of its logical consequences that will be tested for it in the RDF repository, and the number of classes that have at least a known instance in common with C. We used this product to define a function predicting the time it takes to test a candidate axiom. This allows to dynamically time-cap the SPARQL queries launched to compute the score of a candidate axiom by a time out defined as a function of the time predictor of the tested axiom. As a corollary, our second heuristics consists in generating the candidate axioms of the form $C \sqsubseteq D$, by considering the subclasses C in increasing order of time predictor. This enables us to maximize the number of tested and accepted axioms in a given time period.

We evaluated the proposed scoring heuristics on DBpedia by performing two experiments: an explorative scoring of systematically generated subsumption axioms (which are manually evaluated) and an exhaustive scoring of all subsumption axioms in the DBpedia ontology (which should be positively scored). The results of experimental evaluation on the DBpedia dataset clearly indicate that the proposed heuristics is suitable for axiom induction and ontology learning.

Conclusion

The research questions addressed in this chapter, namely the creation of RDF data from heterogeneous Web data, the presentation and transformation of RDF data, the validation of RDF data, and ontology learning from RDF data, all are topical issues in the semantic Web community and the research works described in this chapter should all be continued. Indeed, with the semantic Web adoption and deployment even W3C redirected its activity and broaden it to the general problem of data on the Web, to support data exchanges and flows across formats, sources, models, etc. The work on provenance and dataset description also acknowledge the fact that the variety of sources, processes, algorithms and actors involved is growing and needs to be tracked and studied.

Conclusion and Perspectives

Conclusion

This document provided an overview of my research activity from 2000 to 2016. It has taken place within various local, national or international projects. My research area are knowledge engineering, graph-based and semantic Web-based knowledge representation and reasoning, with the general aim of supporting online epistemic communities in the capitalisation and management of their digital resources and knowledge. The semantic Web can be viewed as a successful reedition, at a giant scale, of a first limited attempt in Artificial Intelligence to develop knowledge based systems. From this point of view, During this 16-year period, I addressed the following three general research questions which are re-actualisation of research questions in Artificial Intelligence: *How should we model digital resources of epistemic communities in order to efficiently manage them?; How should we model community members, social structures and social interactions in order to further improve the support to epistemic communities?; How should we represent knowledge and reason on the semantic Web?*. My contributions have been published in national and international conferences and journals of the relevant scientific communities. They mainly deal with:

(1) Vocabulary-oriented modelling and processing of the knowledge of communities, focusing on the construction of domain vocabularies, the semantic annotation of and reasoning on the digital information resources of epistemic communities, the capture, representation and exploitation of know-how knowledge of epistemic communities, the preservation of cultural heritage;

(2) Vocabulary-oriented modelling and reasoning on communities and community members, focusing on modelling individual profiles or contexts of community members, detecting communities and modelling social structures, reasoning on community members and social structures for searching, browsing, recommending digital resources, and natural language question answering;

(3) Graph-based KRR models for the semantic Web, focusing on taking into account the graph nature of the Web;

(4) Advanced Linked Data processing, dealing with the heterogeneity and distribution of data on the Web, focusing on the production of linked data from the deep Web or any other source and its transformation to other data formats, its presentation to the user and its validation against constraints.

On-going work and perspectives

In the continuation of my ongoing research work, and in line with the objectives of the Wimmics team, I will keep addressing the three general research questions discussed in this document. Considering the tremendous growth of the social Web, I intend to give a growing importance to the social dimension of the Web. Meanwhile, I plan to emphasize research projects in e-Education and Cultural Heritage.

(1) On vocabulary-oriented modelling and managing of digital resources of epistemic communities, I intend to contribute to the development of reference ontologies in e-Education, for various school levels, enabling to annotate and integrate heterogeneous learning resources. Additionnally, I recently initiated a research activity aiming at automatically *creating* learning resources by exploiting the Web of data, and more specifically educational quizzes. This should come in the continuation of my ongoing collaborations with the French Gayatech company on serious games and intelligent quizzes [81] and with the French company Educlever on adaptive and collaborative learning, both for primary and secondary schools, and in the framework of the SIDES 3.0 ANR project on intelligent learning environments for medical students. I also intend to develop my activity initiated in the framework of the Zoomathia project on the construction of a reference vocabulary for digital studies in ancient and mediaeval zoology and on the automatic annotation of textual resources based on knowledge extraction from texts using natural language processing methods. In another domain, the starting collaboration with the SILEX company on linking service providers and clients should provide other use cases to apply similar approaches to annotate textual descriptions of service offers and supplies.

(2) On supporting user interactions, I intend to develop my research activity initiated within the Zoomathia project on the visualisation of data and query results for non expert users enabling them to visually analyse data. This should be based on the use of STTL language for this special need of data transformation. The EduMICS project with the French company Educlever and the SIDES 3.0 project should provide other use cases of data visualisation in e-Education. I also intend to continue developing my research activity initiated within my ongoing collaboration with the French company SynchroNext on NL-based user interactions in question answering systems for *specific* enterprise application domains, enabling to automatically learn *domain-dependent* query patterns, to generate corresponding natural language answer patterns, and to answer complex queries based on linked data. My longer-term programme includes supporting natural language *dialog* with an artificial conversational agent.

(3) On modelling and reasoning on social interactions, I intend to develop research projects in e-Education, dealing with detecting learning communities and supporting learning processes based on the modelling and analysis of the detected social structures. This should come in the continuation of my ongoing collaboration with the French company Educlever on adaptive and collaborative learning. In another domain, the collaboration with the SILEX company on linking service providers and clients should provide other use cases to detect communities of providers and clients and refine the linking accordingly.

(4) On Linked Data processing, I will continue working on all the projects presented in Chapter 4. As a followup to Franck Michel PhD thesis, we intend to consider taking into account SPARQL entailment regimes while querying

non relational data and to address the question of distributing and federating queries to query distributed heterogeneous datasets. In the continuation of our research project on *automatic* vocabulary learning from the Web of data, Andrea Tettamanzi, Fabien Gandon and I are currently developing a theory of OWL axiom testing against RDF facts based on possibility theory. Also, to handle the wide range of OWL axiom types will require us to find specific heuristics allowing to test expressive axioms on large RDF datasets. In the continuation of our work on STTL, Olivier Corby, Fabien Gandon and I are currently working on the definition of a scripting language for the semantic Web, based on SPARQL. Finally, in the continuation of my works on validating data and showing the results of the validation process, my longer term programme includes tracing and explaining automatic reasoning, and ultimately aims to achieve a Web intelligence combining actions from human and artificial agents.

Bibliography

[1] Jean-François Baget, Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, Fabien L. Gandon, Alain Giboin, Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, and Rallou Thomopoulos. Griwes: Generic model and preliminary specifications for a graph-based knowledge representation toolkit. In *16th International Conference on Conceptual Structures, ICCS 2008, Toulouse, France*, volume 5113 of *Lecture Notes in Computer Science*, pages 297–310. Springer, 2008.

[2] Jean-François Baget and Marie-Laure Mugnier. Extensions of simple conceptual graphs: the complexity of rules and constraints. *J. Artif. Intell. Res. (JAIR)*, 16:425–465, 2002.

[3] Samia Beldjoudi. *La Sémantique et l'Effet Communautaire: Enrichissement et Exploitation*. PhD thesis, Université Badji Mokhtar, Annaba, Algeria, 2015.

[4] Samia Beldjoudi, Hassina Seridi, and Catherine Faron-Zucker. Improving tag-based resource recommendation with association rules on folksonomies. In *2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation, SPIM 2011, Bonn, Germany*, CEUR Workshop Proceedings, pages 26–37, 2011.

[5] Khalil Riad Bouzidi. *Aide à la création et à l'exploitation de réglementations basée sur les modèles et techniques du Web sémantique*. PhD thesis, Université Nice Sophia Antipolis, France, 2013.

[6] Khalil Riad Bouzidi, Catherine Faron-Zucker, Bruno Fiès, Olivier Corby, and Nhan Le Thanh. Aide à la rédaction de documents réglementaires dans le domaine du bâtiment. In *23es Journées Francophones d'Ingénierie des Connaissances, IC 2012*, pages 235–250, Paris, France, 2012.

[7] Khalil Riad Bouzidi, Catherine Faron-Zucker, Bruno Fiès, Olivier Corby, and Nhan Le Thanh. Towards a semantic-based approach for modeling regulatory documents in building industry. In *9th European Conference on Product and Process Modelling, ECPPM 2010, Reykjavik, Iceland*, 2012.

[8] Khalil Riad Bouzidi, Catherine Faron-Zucker, Bruno Fiès, and Nhan Le Thanh. An ontological approach for modeling technical standards for compliance checking. In *5th International Conference on Web Reasoning and Rule Systems, RR 2011, Galway, Ireland*, volume 6902 of *Lecture Notes in Computer Science*, pages 244–249. Springer, 2011.

[9] Khalil Riad Bouzidi, Bruno Fiès, Marc Bourdeau, , Catherine Faron-Zucker, and Nhan Le Thanh. Toward a semantic-based approach for the creation of technical regulatory documents in building industry. In *8th European Conference on Product and Process Modelling, ECPPM 2010, Cork, Ireland*, pages 217–222. CRC Press, 2010.

[10] Khalil Riad Bouzidi, Bruno Fiès, Marc Bourdeau, Catherine Faron-Zucker, and Nhan Le Thanh. An ontology for modelling and supporting the process of authoring technical assessments. In *28th International Conference CIB W78 Information Technology for Construction, Sophia Antipolis, France*, 2011.

[11] Khalil Riad Bouzidi, Bruno Fiès, Catherine Faron-Zucker, Alain Zarli, and Nhan Le Thanh. Semantic web approach to ease regulation compliance checking in construction industry. *Future Internet*, 4(3):830–851, 2012.

[12] Christian Brel. *Composition d'applications multi-modèles dirigée par la composition des interfaces graphiques*. PhD thesis, Université Nice Sophia Antipolis [UNS], 2013.

[13] Christian Brel, Anne-Marie Pinna Dery, Catherine Faron-Zucker, Philippe Renevier-Gonin, and Michel Riveill. Ontocompo: An ontology-based interactive system to compose applications. In *7th International Conference on Web Information Systems and Technologies, WEBIST 2011, Noordwijkerhout, The Netherlands*, pages 322–327. SciTePress, 2011.

[14] Christian Brel, Philippe Renevier-Gonin, Audrey Occello, Anne-Marie Pinna Dery, Catherine Faron-Zucker, and Michel Riveill. Ontocompo: An ontology-based interactive system to compose applications. In *Third International Conference on Human Centred Software Engineering, HCSE 2010, Reykjavik, Iceland*, volume 6409 of *Lecture Notes in Computer Science*, pages 198–205. Springer, 2010.

[15] Michel Buffa and Catherine Faron-Zucker. Ontology-based access rights management. In *Advances in Knowledge Discovery and Management - Volume 2 [Best of EGC 2010, Hammamet, Tunisie]*, volume 398 of *Studies in Computational Intelligence*, pages 49–61. Springer, 2010.

[16] Michel Buffa, Catherine Faron Zucker, Thierry Bergeron, and Hatim Aouzal. Semantic Web Technologies for improving remote visits of museums, using a mobile robot. In *ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan*, volume 1690 of *CEUR Workshop Proceedings*, 2016.

[17] Michel Buffa, Fabien L. Gandon, Guillaume Erétéo, Peter Sander, and Catherine Faron. Sweetwiki: A semantic wiki. *Journal of Web Semantics*, 6(1):84–97, 2008.

[18] Elena Cabrio, Catherine Faron-Zucker, Fabien Gandon, Amine Hallili, and Andrea G. B. Tettamanzi. Answering N-Relation Natural Language Questions in the Commercial Domain. In *The IEEE/WIC/ACM International Conference on Web Intelligence*, Singapore, Singapore, 2015.

[19] Elena Cabrio, Sara Tonelli, Serena Villata, Ahmed Missaoui, and Catherine Faron-Zucker. Semantic Linking to Enrich Small Artwork Collections: Experiences with Archivio di Nuova Scrittura. In *Digital Humanities e beni culturali: quale relazione?, Torino, Italy*, 2015.

[20] Cécile Callou, Franck Michel, Catherine Faron-Zucker, Chloé Martin, and Johan Montagnat. Towards a shared reference thesaurus for studies on history of zoology, archaeozoology and conservation biology. In *1st International Workshop Semantic Web for Scientific Heritage at the 12th ESWC 2015 Conference, Portorož, Slovenia, 2015*, volume 1364 of *CEUR Workshop Proceedings*, pages 15–22, 2015.

[21] Michel Chein and Marie-Laure Mugnier. *Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer, 2009.

[22] Michel Chein, Marie-Laure Mugnier, and Geneviève Simonet. Nested graphs: A graph-based knowledge representation model with FOL semantics. In *6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy*, pages 524–535. Morgan Kaufmann, 1998.

[23] Olivier Corby, Rose Dieng, and Cédric Hébert. A conceptual graph model for W3C resource description framework. In *8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany*, volume 1867 of *Lecture Notes in Computer Science*, pages 468–482. Springer, 2000.

[24] Olivier Corby, Rose Dieng-Kuntz, and Catherine Faron-Zucker. Querying the semantic web with corese search engine. In *16th Eureopean Conference on Artificial Intelligence, ECAI 2004, Valencia, Spain*, pages 705–709. IOS Press, 2004.

[25] Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, and Fabien L. Gandon. Searching the semantic web: Approximate query processing based on ontologies. *IEEE Intelligent Systems*, 21(1):20–27, 2006.

[26] Olivier Corby, Caroline Domerg, Juliette Fabre, Catherine Faron-Zucker, Isabelle Mirbel, Vincent Négre, and Pascal Neveu. Using ontologies for R functions management. In *Book of contributed abstracts of the R User Conference, UseR! 2010, Gaithersburg, Maryland, USA*, page 113, 2010.

[27] Olivier Corby and Catherine Faron-Zucker. A corporate semantic web engine. In *International WWW Workshop on Real World RDF and Semantic Web Applications, Honolulu, Hawaii, USA*, 2002.

[28] Olivier Corby and Catherine Faron-Zucker. Implementation of SPARQL query language based on graph homomorphism. In *15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK*, volume 4604 of *Lecture Notes in Computer Science*, pages 472–475. Springer, 2007.

[29] Olivier Corby and Catherine Faron-Zucker. RDF/SPARQL design pattern for contextual metadata. In *IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007, Silicon Valley, CA, USA*, pages 470–473. IEEE Computer Society, 2007.

[30] Olivier Corby and Catherine Faron-Zucker. The KGRAM abstract machine for knowledge graph querying. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Toronto, Canada*, pages 338–341. IEEE Computer Society, 2010.

[31] Olivier Corby and Catherine Faron-Zucker. La machine abstraite KGRAM et son langage GRAAL pour l'interrogation de graphes de connaissances. In *Langages et modèles à objets, Pau, France*, 2010.

[32] Olivier Corby and Catherine Faron-Zucker. STTL - A sparql-based transformation language for RDF. In *11th International Conference on Web Information Systems and Technologies, WEBIST 2015, Lisbon, Portugal*, pages 466–476. SciTePress, 2015.

[33] Olivier Corby and Catherine Faron Zucker. Un navigateur pour les données liées du Web. In *26es Journées francophones d'Ingénierie des Connaissances, IC 2015, Rennes, France*, 2015.

[34] Olivier Corby and Catherine Faron-Zucker. Un language et un serveur de transformation de graphes pour le Web de données. *Revue d'Intelligence Artificielle*, 30(5), 2016.

[35] Olivier Corby, Catherine Faron-Zucker, and Fabien Gandon. A Generic RDF Transformation Software and its Application to an Online Translation Service for Common Languages of Linked Data. In *14th International Semantic Web Conference, ISWC 2015, Bethlehem, USA*, 2015.

[36] Olivier Corby, Catherine Faron-Zucker, and Raphaël Gazzotti. Validating ontologies against OWL 2 profiles with the SPARQL template transformation language. In *10th International Conference on Web Reasoning and Rule Systems, RR 2016, Aberdeen, UK*, volume 9898 of *Lecture Notes in Computer Science*, pages 39–45. Springer, 2016.

[37] Olivier Corby, Catherine Faron-Zucker, and Isabelle Mirbel. Démarches sémantiques de recherche d'information sur le web. In *20es Journées Francophones d'Ingénierie des Connaissances, Hammamet, Tunisia*, pages 289–300. PUG, 2009.

[38] Olivier Corby, Catherine Faron-Zucker, and Isabelle Mirbel. Implementation of intention-driven search processes by SPARQL queries. In *11th International Conference on Enterprise Information Systems, ICEIS 2009, Milan, Italy*, pages 339–342, 2009.

[39] Olivier Corby, Alban Gaignard, Catherine Faron-Zucker, and Johan Montagnat. KGRAM versatile inference and query engine for the web of linked data. In *IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012, Macau, China*, pages 121–128. IEEE Computer Society, 2012.

[40] Pascal Coupey and Catherine Faron. Towards correspondence between conceptual graphs and description logics. In *6th International Conference on Conceptual Structures, ICCS '98, Montpellier, France*, volume 1453 of *Lecture Notes in Computer Science*, pages 165–178. Springer, 1998.

[41] Madalina Croitoru and Ernesto Compatangelo. A combinatorial approach to conceptual graph projection checking. In *Twenty-fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK*, pages 130–143. Springer, 2004.

[42] Sylvain Dehors. *Exploiting Semantic Web and Knowledge Management Technologies for E-learning.* PhD thesis, Université Nice Sophia Antipolis, France, 2007.

[43] Sylvain Dehors and Catherine Faron-Zucker. QBLS: A Semantic Web Based Learning System. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA 2006, Orlando, Florida, USA*, 2006.

[44] Sylvain Dehors and Catherine Faron-Zucker. Reusing learning resources based on semantic web technologies. In *6th IEEE International Conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, The Netherlands*, pages 859–863. IEEE Computer Society, 2006.

[45] Sylvain Dehors, Catherine Faron-Zucker, and Rose Dieng-Kuntz. QBLS: Semantic Web Technology for E-Learning in Practice. In *15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006, Podebrady, Czech Republic, Poster & Demo Proc.*, pages 7–8, 2006.

[46] Sylvain Dehors, Catherine Faron-Zucker, Jean-Paul Stromboni, and Alain Giboin. Des annotations sémantiques pour apprendre. QBLS : Modélisation et expérimentation. In *Journée Web sémantique pour le e-Learning, PFIA 2005, Nice, France*, pages 15–30, 2005.

[47] Alexandre Delteil. *Représentation et apprentissage de concepts et d'ontologies pour le web sémantique.* PhD thesis, Université Nice Sophia Antipolis, France, 2002.

[48] Alexandre Delteil and Catherine Faron. A graph-based knowledge representation language for concept description. In *15th European Conference on Artificial Intelligence, ECAI 2002, Lyon, France*, pages 297–301. IOS Press, 2002.

[49] Alexandre Delteil, Catherine Faron, and Rose Dieng. Learning Ontologies from RDF Annotations. In *IJCAI 2001 Workshop on Ontologies and Information Sharing, Seattle, USA*, volume 47 of *CEUR Workshop Proceedings*, pages 147–156, 2001.

[50] Alexandre Delteil, Catherine Faron, and Rose Dieng. Building Concept Lattices by Learning Concepts from RDF Graphs Annotating Web Documents. In *10th International Conference on Conceptual Structures, ICCS 2002, Borovets, Bulgaria*, volume 2393 of *Lecture Notes in Computer Science*, pages 191–204. Springer, 2002.

[51] Alexandre Delteil, Catherine Faron-Zucker, and Rose Dieng. Extension of RDFS based on the cgs formalisms. In *9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA*, volume 2120 of *Lecture Notes in Computer Science*, pages 275–289. Springer, 2001.

[52] Alexandre Delteil, Catherine Faron-Zucker, and Rose Dieng. Le modèle des graphes conceptuels pour le web sémantique extensions de RDF et RDFS basées sur le modèle des graphes conceptuels. *L'OBJET*, 9(3):95–122, 2003.

[53] Molka Dhouib, Catherine Faron Zucker, Arnaud Zucker, Olivier Corby, Catherine Jacquemard, Isabelle Draelants, and Pierre-Yves Buard. Transformation et visualisation de données RDF à partir d'un corpus annoté de textes médiévaux latins. In *26e conférence francophone sur l'Interaction Homme-Machine (IHM'14), Lille, France, atelier Visualisation d'information, fouille visuelle de données et nouveaux challenges en Big data et Humanités numériques*, pages 63–68, 2014.

[54] Rose Dieng-Kuntz, Monique Grandbastien, and Danièle Hérin, editors. *Actes de la Journée Web sémantique pour le e-Learning, PFIA 2005, Nice, France*, 2005.

[55] Guillaume Erétéo, Michel Buffa, Fabien Gandon, and Olivier Corby. Analysis of a real online social network using semantic web frameworks. In *8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA*, volume 5823 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 2009.

[56] Catherine Faron-Zucker, Irene Pajón Leyra, Konstantina Poulida, and Andrea G. B. Tettamanzi. Semantic categorization of segments of ancient and mediaeval zoological texts. In *2nd International Workshop Semantic Web for Scientific Heritage at the 13th ESWC 2016 Conference, Heraklion, Greece*, volume 1595 of *CEUR Workshop Proceedings*, pages 59–68, 2016.

[57] Catherine Faron-Zucker, Anastasiya Yurchyshyna, Nhan Le Thanh, and Celson Lima. Une approche ontologique pour automatiser le contrôle de conformité dans le domaine du bâtiment. In *8èmes journées Extraction et Gestion des Connaissances, EGC 2008, Sophia-Antipolis, France*, volume RNTI-E-11 of *Revue des Nouvelles Technologies de l'Information*, pages 115–120. Cépaduès-Éditions, 2008.

[58] Alban Gaignard. *Distributed knowledge sharing and production through collaborative e-Science platforms*. PhD thesis, Université Nice Sophia Antipolis, 2013.

[59] Alban Gaignard, Johan Montagnat, Catherine Faron-Zucker, and Olivier Corby. Fédération multi-sources en neurosciences : intégration de données relationnelles et sémantiques. In *Atelier IC pour l'Interopérabilité Sémantique dans les applications en e-Santé*, Paris, France, 2012.

[60] Alban Gaignard, Johan Montagnat, Catherine Faron-Zucker, and Olivier Corby. Semantic Federation of Distributed Neurodata. In *MICCAI Workshop on Data- and Compute-Intensive Clinical and Translational Imaging Applications*, pages 41–50, Nice, France, 2012.

[61] Fabien Gandon. *Distributed Artificial Intelligence And Knowledge Management: Ontologies And Multi-Agent Systems For A Corporate Semantic Web*. PhD thesis, Université Nice Sophia Antipolis, 2002.

[62] Fabien Gandon. *RDF Graphs and their manipulation for knowledge management.* Habilitation à diriger des recherches, Université Nice Sophia Antipolis, 2008.

[63] Fabien Gandon, Michel Buffa, Elena Cabrio, Olivier Corby, Catherine Faron-Zucker, Alain Giboin, Nhan Le Thanh, Isabelle Mirbel, Peter Sander, Andrea Tettamanzi, and Serena Villata. Challenges in bridging social semantics and formal semantics on the web. In *15th International Conference on Enterprise Information Systems, ICEIS 2013, Angers, France*, volume 190 of *Lecture Notes in Business Information Processing*, pages 3–15. Springer, 2013.

[64] Jennifer Golbeck and James A. Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *14th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2004, Whittlebury Hall, UK*, pages 116–131, 2004.

[65] Tom Gruber. Where the social web meets the semantic web. In *5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA*, page 994, 2006.

[66] Amine Hallili, Elena Cabrio, and Catherine Faron-Zucker. QALM: a benchmark for question answering over linked merchant websites data. In *ISWC 2014 Posters & Demonstrations Track, a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy*, volume 1272 of *CEUR Workshop Proceedings*, pages 389–392, 2014.

[67] Gilles Kahn. Natural semantics. In *4th Annual Symposium on Theoretical Aspects of Computer Science, STACS 87, Passau, Germany*, volume 247 of *Lecture Notes in Computer Science*, pages 22–39. Springer, 1987.

[68] Michel Leclère, Francky Trichet, Olivier Corby, and Catherine Faron-Zucker, editors. *Actes de la journée Raisonner le Web Sémantique avec des Graphes, Nice, France*, 2005.

[69] Brian McBride. Jena: Implementing the RDF model and syntax specification. In *Second International Workshop on the Semantic Web, SemWeb 2001, Hongkong, China*, CEUR Workshop Proceedings, 2001.

[70] Zide Meng. *Temporal and semantic analysis of richly typed social networks from user-generated content sites on the Web.* PhD thesis, Université Nice Sophia Antipolis [UNS], 2016.

[71] Zide Meng, Fabien Gandon, and Catherine Faron-Zucker. Simplified labeling of overlapping communities of interest in question-and-answer sites. In *The 2015 IEEE/WIC/ACM International Conference on Web Intelligence*, Singapore, Singapore, 2015.

[72] Zide Meng, Fabien L. Gandon, and Catherine Faron-Zucker. QASM: a q&a social media system based on social semantic. In *ISWC 2014 Posters & Demonstrations Track, a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy*, volume 1272 of *CEUR Workshop Proceedings*, pages 333–336, 2014.

[73] Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, and Ge Song. Empirical study on overlapping community detection in question and answer sites. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China*, pages 344–348. IEEE, 2014.

[74] Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, and Ge Song. Detecting topics and overlapping communities in question and answer sites. *Social Network Analysis Mining*, 5(1):27:1–27:17, 2015.

[75] Franck Michel. *Integrating Heterogeneous Data Sources in the Web of Data*. PhD thesis, Université Nice Sophia Antipolis [UNS], 2017.

[76] Franck Michel, Loïc Djimenou, Catherine Faron-Zucker, and Johan Montagnat. Translation of relational and non-relational databases into RDF with xR2RML. In *11th International Conference on Web Information Systems and Technologies, WEBIST 2015, Lisbon, Portugal*, pages 443–454. SciTePress, 2015.

[77] Franck Michel, Catherine Faron-Zucker, and Johan Montagnat. A Generic Mapping-Based Query Translation from SPARQL to Various Target Database Query Languages. In *12th International Conference on Web Information Systems and Technologies, WEBIST 2016, Roma, Italy*, volume 2, pages 147–158. SciTePress, 2016.

[78] Franck Michel, Catherine Faron-Zucker, and Johan Montagnat. A Mapping-based Method to Query MongoDB Documents with SPARQL. In *27th International Conference on Database and Expert Systems Applications, DEXA 2016, Porto, Portugal*, volume 9828 of *Lecture Notes in Computer Science*, pages 52–67. Springer, 2016.

[79] Pascal Neveu, Caroline Domerg, Juliette Fabre, Vincent Négre, Emilie Gennari, Anne Tireau, Olivier Corby, Catherine Faron-Zucker, and Isabelle Mirbel. Using ontologies of software: Example of R functions management. In *3rd International Workshop on Resource Discovery, RED 2010, Paris, France, Revised Selected Papers*, volume 6799 of *Lecture Notes in Computer Science*, pages 43–56. Springer, 2010.

[80] Irene Pajón Leyra, Arnaud Zucker, and Catherine Faron Zucker. Thezoo: un thesaurus de zoologie ancienne et médiévale pour l'annotation de sources de données hétérogènes. *Archivum Latinitatis Medii Aevi*, 73:321–342, 2015.

[81] Oscar Rodriguez Rocha and Catherine Faron-Zucker. An Ontology to Create Linked Data Driven Serious Games. In *ISWC 2015 Workshop on LINKed EDucation, LINKED 2015*, Bethlehem, Pennsylvania, United States, 2015.

[82] Eric Salvat and Marie-Laure Mugnier. Sound and complete forward and backward chainingd of graph rules. In *4th International Conference on Conceptual Structures, ICCS '96, Sydney, Australia*, volume 1115 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 1996.

[83] Oumy Seye. *Partage et réutilisation de règles sur le Web de données.* PhD thesis, Université Nice Sophia Antipolis, France & Université Gaston Berger, Sénégal, 2014.

[84] Oumy Seye, Catherine Faron-Zucker, Olivier Corby, and Corentin Follenfant. Bridging the gap between rif and sparql: Implementation of a rif dialect with a sparql rule engine. In *ECAI 2012 Workshop Artificial Intelligence meets the Web of Data, AImWD 2012, Montpellier, France*, 2012.

[85] Oumy Seye, Catherine Faron-Zucker, Olivier Corby, and Alban Gaignard. Publication, partage et réutilisation de règles sur le Web de données. In *25es Journées francophones d'Ingénierie des Connaissances, IC 2014, Clermont-Ferrand, France*, pages 237–248, 2014.

[86] Derek Sleeman and John Seely Brown, editors. *Intelligent Tutoring Systems.* Academic Press, 1982.

[87] John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley, 1984.

[88] John F. Sowa. Conceptual graphs: Draft proposed american national standard. In *7th International Conference on Conceptual Structures, ICCS '99, Blacksburg, Virginia, USA*, volume 1640 of *Lecture Notes in Computer Science*, pages 1–65. Springer, 1999.

[89] Arthur Stutt and Enrico Motta. Semantic webs for learning: A vision and its realization. In *14th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2004, Whittlebury Hall, UK*, pages 132–143, 2004.

[90] Andrea G. B. Tettamanzi, Catherine Faron Zucker, and Fabien Gandon. Dynamically Time-Capped Possibilistic Testing of SubClassOf Axioms Against RDF Data to Enrich Schemas. In *8th International Conference on Knowledge Capture, K-CAP 2015*, Palisades, NY, United States, October 2015.

[91] Andrea G. B. Tettamanzi, Catherine Faron-Zucker, and Fabien L. Gandon. Testing OWL Axioms against RDF Facts: A Possibilistic Approach. In *19th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2014, Linköping, Sweden*, volume 8876 of *Lecture Notes in Computer Science*, pages 519–530. Springer, 2014.

[92] Molka Tounsi, Catherine Faron-Zucker, Arnaud Zucker, Serena Villata, and Elena Cabrio. Studying the history of pre-modern zoology with linked data and vocabularies. In *1st International Workshop Semantic Web for Scientific Heritage at the 12th ESWC 2015 Conference, Portorož, Slovenia*, volume 1364 of *CEUR Workshop Proceedings*, pages 7–14, 2015.

[93] Mike Uschold and Michael Gruninger. Ontologies: principles, methods and applications. *Knowledge Eng. Review*, 11(2):93–136, 1996.

[94] Serena Villata, Luca Costabello, Fabien Gandon, Catherine Faron Zucker, and Michel Buffa. Social Semantic Network-Based Access Control. In *Security and Privacy Preserving in Social Networks*, Lecture Notes in Social Networks. Springer, 2013.

[95] Amel Yessad. *Construction d'un environnement pédagogique adaptatif basé sur les modèles et techniques du Web sémantique*. PhD thesis, Université Badji Mokhtar, Annaba, Algeria, 2009.

[96] Amel Yessad, Catherine Faron-Zucker, Rose Dieng-Kuntz, and Med Tayeb Laskri. Adaptive learning organizer for web-based education. *International Journal of Web-Based Learning and Teaching Technologies, IJWLTT*, 3(4):57–73, 2008.

[97] Amel Yessad, Catherine Faron-Zucker, Rose Dieng-Kuntz, and Med Tayeb Laskri. Ontology-based semantic relatedness for detecting the relevance of learning resources. *Interactive Learning Environments*, 19(1):63–80, 2011.

[98] Anastasiya Yurchyshyna. *Modélisation du contrôle de conformité en construction : une approche ontologique*. PhD thesis, Université Nice Sophia Antipolis, France, 2009.

[99] Anastasiya Yurchyshyna, Catherine Faron-Zucker, Isabelle Mirbel, Birahim Sall, Nhan Le Thanh, and Alain Zarli. Une approche ontologique pour formaliser la connaissance experte dans le modèle du contrôle de conformité en construction. In *19es Journées Francophones d'Ingénierie des Connaissances, IC 2008*, pages 49–60, Nancy, France, 2008.

[100] Anastasiya Yurchyshyna, Catherine Faron-Zucker, Nhan Le Thanh, and Alain Zarli. Towards an ontology based approach for conformance checking modeling in construction. In *24th International Conference CIB W78 Information Technology for Construction, Maribor, Slovenia*, 2007.

[101] Anastasiya Yurchyshyna, Catherine Faron-Zucker, Nhan Le Thanh, and Alain Zarli. Towards an ontology-enabled approach for modeling the process of conformity checking in construction. In *Forum at the CAiSE'08 Conference, Montpellier, France*, volume 344 of *CEUR Workshop Proceedings*, pages 21–24, 2008.

[102] Anastasiya Yurchyshyna, Catherine Faron-Zucker, Nhan Le Thanh, and Alain Zarli. Adaptation of the domain ontology for different user profiles: Application to conformity checking in construction. In *5th International Conference on Web Information Systems and Technologies, WEBIST 2009, Lisbon, Portugal, Revised Selected Papers*, volume 45 of *Lecture Notes in Business Information Processing*, pages 128–141. Springer, 2009.

[103] Anastasiya Yurchyshyna, Catherine Faron-Zucker, Nhan Le Thanh, and Alain Zarli. Knowledge capitalisation and organisation for conformance checking model in construction. *International Journal of Knowledge Engineering and Soft Data Paradigms, IJKESDP*, 2(1):15–32, 2010.